PRIJS ƒ 7.25/Bfr. 135

Commodore INFO

ONAFHANKELIJK BLAD VOOR COMMODORE GEBRUIKERS

JAARGANG 5, NO. 5, augustus 1988



Commodore Info

Verschijnt 8x per jaar

Jaarg.5, no.5, aug. 1988

Uitgave:
Sala Communications/SAC
Uitgever:
Vic Sharfman
Redactie:
Ir. L. Sala hoofdredacteur
J. Bodzinga adj. hoofdredacteur
drs. J. Boers eindredakteur
drs. M. de Rooij, drs. H. Zoete,
H. Smeenk, drs. U. Schuurmans,
R. Goudriaan, B. Munniksma,
P.C. Broekhuizen.
Redactiesecretariaat:
R. van Zalingen
Tel. 020-228871
Strip: Bert Tier
Illustraties: Ben van Mierlo
Advertentie-exploitatie:
Ing. V. Sala, Ing. B. Sala,
D. van Vlijmen
Weesperstraat 103
1018 VN Amsterdam
tel. 020-273198
Redactie adres:
Postbus 43048
1009 ZA Amsterdam
tel. 020-228871
Listingtelefoon:
(ma: 17.00-21.00) 02155-25162
Abonnementen en administratie:
Nicole Balke en Marjo Jansen
Postbus 43048
1009 ZA Amsterdam
tel. 020-248006
Vragen betreffende abonnementen ontvangen wij bij voorkeur schriftelijk, met meesturen van het omslagetiket.
Abonnement:
Voor 8 nummers ƒ 47,50 of Bfr. 975 per jaar. Betaling op giro 1585491 t.n.v. SAC/Commodore-Info. Oude nummers kunt U alleen krijgen bij vooruitbetaling van ƒ 6,75 op de bovenstaande rekening. Ook telefonische opgave voor een abonnement is mogelijk. Bel GRATIS 06-02242222 (teleservice), elke dag tot 20.20 uur (dus ook in het weekend). België: 115555, dagelijks tot 22.00 uur. Deze telefoonnummers zijn alleen bedoeld voor opgave van NIEUWE abonnementen.
Opzegging dient schriftelijk te geschieden uiterlijk twee maanden voor de aanvang van een nieuwe abonnementsperiode van een jaar.

Druk: NDB, Zoeterwoude
Distributie:
In Nederland: Betapress, Gilze
In België: AMP, Brussel

ISSN: 0169-3085

Inhoud van dit nummer

Omslagfoto: Gould Electronic

Redaktioneel

Onze vaste lezers van Commodore Info zullen wel wat verbaasd zijn, als ze het blad doorbladeren. Want er is wel wat veranderd in de vormgeving. Terwijl den meeste mensen op het strand lagen, hebben wij geprobeerd de vormgeving van ons blad wat te 'updaten'. Dat was nodig, omdat er naast veel 16 en 64 gebruikers ook een grote schare 128 en Amiga aanhangers is, die in hun lijfblad natuurlijk ook aan hun trekken willen komen. Wij hopen, dat deze nieuwe vorm de lezers nog meer leesplezier geeft, en dat de vakantiestemming door dit nummer nog wat zal blijven hangen.
En mocht u helaas wat zijn weggespoeld op de camping, dan kunt u altijd binnenkort nog een heerlijk uitje gaan maken naar de Firato, die dit jaar haar tenten weer zal opslaan in de RAI (26 aug. - 4 sep.) We hoeven onze lezers niet meer te vertellen, dat er ook dit jaar weer veel computernieuws zal komen. Vooral de koppeling met videosystemen zal een populair item worden, als we de berichten mogen geloven.

De redaktie

Datasoft Adventures

Het Amerikaanse softwarehuis Datasoft maakt al jaren allerlei software, meest spelletjes en eenvoudige productivity pakketjes. Sinds enige tijd is Electronic Arts de distributeur van de Datasoft reeks. Dat betekent, dat ook in ons land via bijvoorbeeld Computer Collectief dergelijke software is aan te schaffen.
Datasoft heeft een aantal nieuwe strategische adventures uitgebracht. Onder die noemer vallen de spelletjes, die naast wat actie ook strategisch inzicht vragen en een planning- en actiefase kennen. Het gaat daarmee verder dan de klassieke Adventure, dat is eigenlijk niet meer dan een soort doolhofspel met een aantal basistrucs, zoals het meenemen van bepaalde voorwerpen en het stellen van de juiste vragen. Voor strategische adventures kiest men meestal een ruimte-scenario of er wordt een historische veldslag nagebootst. Daarmee wordt dan een meestal matig gecompliceerde omgeving gemaakt, waarin men als ware het een 'computer'-boek, op avontuur kan gaan.
Het spel De Rubicon Alliance is gesitueerd in de toekomst van de interplanetaire ruimtevaart. De Rubicon Alliance is een verbond tussen de acht planeten van het Hyturian sterrestelsel. Die alliantie verzekert de vrede maar is ook de beheerder van een anti-materie krachtveld, dat 'The Rubicon' heet.

Winst Commodore Int.

Commodore International heeft in de eerste negen maanden van het fiscale jaar 1987/888 een winst van 43 miljoen dollar gerealiseerd. Dit betekent een verbetering van de resultaten ten opzichte van het vorige jaar, die met name veroorzaakt wordt door een verdere stijging van de verkopen van de Amiga. De Amiga mmaakt op het moment wereldwijd 45 % van de totale omzet van Commodore uit. In de periode van het jaar ervoor was dat nog slechts 13 %. Er zijn op dit moment al meer dan een half miljoen Amiga's verkocht in de wereld.

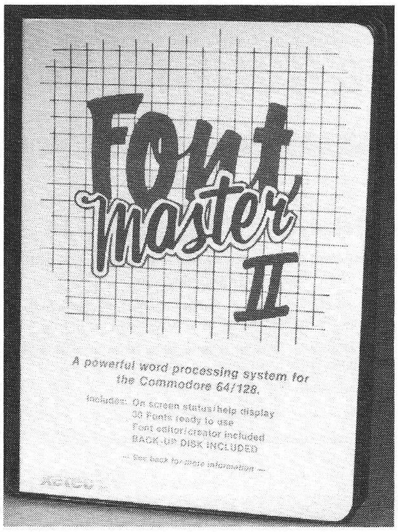
Bard's Tale III: Thief of Fate

Het derde deel in de serie fanatasie-spelletjes gebaseerd op het de middeleeuwen is uit. Na het tweede deel *The Destiny Knight* is dit nog sneller en biedt meer magische mogelijkheden. In deel drie moet je op zoek naar de tovenaar Tarjan, die aan het eind van deel twee de stad Skara Brae heeft vernietigd. De hoofdpersoon kan opgebouwd worden uit een mix van zowel mannelijke als vrouwelijke karakter(trekje)s die ertoe moeten leiden dat de persoonlijkheid sterk genoeg is om de tovenaar te verslaan.



The Bard's Tale III

Fontmaster II



Zowel voor de 64 als de 128 is er een nieuwe versie van Fontmaster, de tekstverwerkers met uitgebreide print- en lettertype mogelijkheden. Fontmaster II is de 64/128 uitvoering die 30 kant-en-klare lettertypes bevat, een lettertype-maker en een back-up diskette. De tekstverwerker heeft de mogeiljkheid om andere talen ook het scherm te tonen, diverse functies als onderstrepen zijn op het scherm zichtbaar en voor text-bewerking zijn meer dan 65 commando's beschikbaar.
Fontmaster 128 bevat daarnaast nog een spellingcontrole, een diskette voor vreemde talen, 45 kant-en-klare lettertypes en uitgebreide printerbesturing.

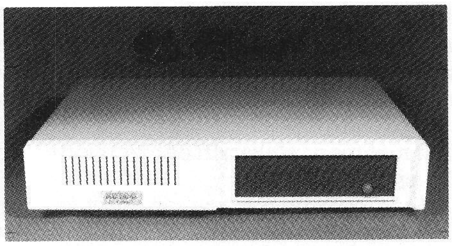
Football Manager II

Als vervolg op het - vooral in Engeland populaire - Football Manager, heeft Prism Leisure nu nummer 2 uitgebracht. De graphics zijn sterk verbeterd, terwijl er ook een aantal extra features zijn toegevoegd. Je kunt nu je eigen (Engelse) team samenstellen, trainingfaciliteiten gebruiken om

het nivo van het team te verhogen, en een replay-mogelijkheid laat je de laatste actie nog eens rustig herhalen. Voor alle spelers met een snelle joystick en matig voetenwerk een kans voor open doel.
Het spel is verkrijgbaar voor C 64, 128 en Amiga's. Voor PC en Amstrad bestaat er eveneens een versie.

Dream Warrior

Als de Dream Warrior in het gelijknamige spel van US Gold bent u de vierde 'Asman'. Het spel valt onder de noemer 'psychologische oorlogsvoering' waarin strategisch inzicht, moeilijke speluitvoering en snelle actie hand in had gaan. Het spel is gesitueerd in in verre toekomst van een onbekend universum, waarin oorlogen slechts in dromen worden uitgevochten, maar dan wel in nachtmeries. De macht is niet langer in handen van regeringen maar van gigantische ondernemingen, en deze ondernemingen worden weer gecontroleerd door Focus.Als geleerde op het gebied van Astrale kennis moet je proberen collega's te bevrijden uit de psychologische gevangeschap van Focus, om uiteindelijk Focus te kunnen vernietigen.
Dream Warrior is er voor de C-64 en IBM PC.



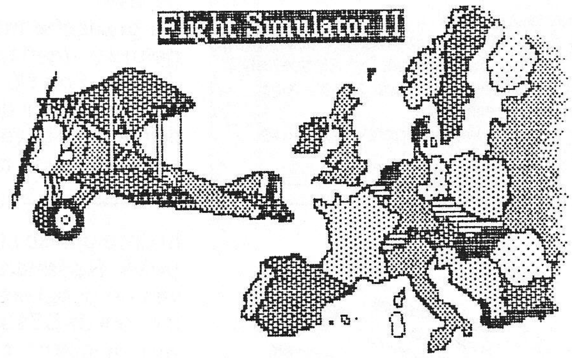
Lt. Kernal Hard Disk Drive

Lt. Kernal C-64 harddisk

Een harddisk voor de C-64 en 128 waarvoor helaas nog geen leverancier in Nederland bekend is. Daardoor weten we slechts een klein aantal gegevens van deze drive. De Lt. Kernal is een 20 MB harddisk in eigen behuizing, met ingebouwde CP/M commando's. Er kunnen tot tien logische drives worden gespecificeerd op de harddisk, terwijl er tevens backup en restore faciliteiten in zitten.

FlightSimulator West Europa Tour

Voor de "piloten" onder ons die wel eens een vlucht maken met een flightsimulator is er goed nieuws. Sinds kort is de "West Europe Tour" diskette van SubLogic in de handel verkrijgbaar. Voor deze scenery-disk is het echter wel noodzakelijk dat je over het programma Flight Simulator II beschikt.



Met deze simulator vliegt de "piloot" in een Piper Archer II, een eenmotorig vliegtuigje, van het ene naar het andere vliegveld. Tot nu toe waren, naast de meegeleverde gebieden, slechts een paar andere Amerikaanse scenery-diskettes verkrijgbaar. Met de komst van de West Europe Tour disk kun je nu ook een vlucht maken in een voor ons enigszins bekendere omgeving. De scenery-disk wordt geladen via de "editor" van het flight simulator programma. Hiermee kunnen de coordinaten worden opgegeven, die corresponderen met de vele op diskette meegeleverde vliegvelden. Ook de natuurverschijnselen kunnen in deze editor worden opgegeven, zoals windsnelheid, de tijd, de jaargetijden, etc. Deze factoren hebben een duidelijk effect op de prestaties van het vliegen op zich. De gebieden welke beschikbaar zijn op de West Europa Tour zijn Zuid-Engeland, Noord-Frankrijk en Zuid Duitsland. Deze gebieden worden op de meegeleverde kaarten uitvoerig getoond. Op deze kaarten zijn diverse vluchtgegevens vermeld, zoals vlieghoogtes, koersen, noodzakelijke communicatiegegevens en de contouren van de belangrijkste vliegvelden. Het vliegen op zich is gelijk aan de vluchten die met het basisprogramma kunnen worden gemaakt. Er is echter een verschil. Met een scenery-disk kan je geen brandstof laden. De brandstoftanks worden automatisch bijgevuld op het moment dat men de editor verlaat. Vele vliegvelden en een paar verrassingen kun je terugvinden in de gebieden, waar je overheen vliegt. De Europa Tour diskette munt uit door z'n gedetailleerde weergave van objecten in een gebied. Ook deze keer is bij het ontwikkelen van deze aanvulling gebruik gemaakt van documentatie die door echte vliegers wordt gebruikt tijdens het navigeren. Gedetailleerde vliegvelden in Zuid-Engeland zijn Bournemouth Cardiff, Heathrow, Luton en Norwich. Zeker de moeite waard is het vliegen over London, waarbij zeer natuurgetrouw enkele bezienswaardigheden worden getoond. Zo kun je mits je niet te hoog of te laag vliegt de Westminster Abbey, de Tower Bridge, Buckingham Pallace, enz. aanschouwen. Wanneer je langs de kust van Engeland vliegt krijg je vanzelf de vuurtorens in zicht. Enkele franse objecten, die het bekijken waard zijn vanuit de cockpit, zijn de Arc de Triomph, de Eiffeltoren, de Notre Dame en nog veel meer. Ook het Zuid-Duitse gebied heeft fraaie objecten te tonen. Maar dit is nog niet alles. Je kunt ook een heldenstaaltje uithalen en proberen te landen op het Rode Plein in Moskou, zonder daarna opgepakt te worden. Of wat dacht je van een rondje Nurnberg-Helsinki.
De Europa Tour diskette is zeker de moeite waard. Er is veel zorg en aandacht aan besteed om een zo realistische mogelijke simulatie te maken. Zeker voor diegenen, die op de reeds bestaande scenery-disks zijn uitgekeken, is de West Europe Tour een welkome afwisseling. Het pakket wordt geleverd met een diskette, 4 vluchtkaarten en aanvullingsbladen betreffende specifieke instructies. De prijs bedraagt ƒ 59,00 en het is verkrijgbaar bij Computer Collectief te Amsterdam. SubLogic's scenery-disk West Europa Tour is, naast de Commodore 64 en 128, tevens verkrijgbaar voor de Amiga, IBM PC en Atari ST.

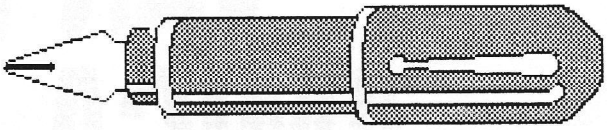
Soft Wir-War

Bert Tier & Wouter Meijer



Onder de titel GEOS Info zullen Bert Venema en Peter Boncz een vaste GEOS rubriek gaan verzorgen. Het is de bedoeling dat ook lezers met tips en trucs terecht kunnen in deze rubriek, dus schrijf gerust met al uw vragen en opmerkingen.

GEOS Info



Directory in GEOS

Het is soms irritant, dat de complete directory van een GEOS-disk niet te zien is vanuit BASIC. Files die namelijk in de border gezet zijn, worden uit de directory gehaald en in een speciale sector op de disk gezet.

Met behulp van BASIC krijg je dus geen goed beeld van de inhoud op een disk. De hierbij geplaatste listing kan daarbij misschien helpen. Het programma leest in de eerste sector van de directory (track 18 sector 0) de bytes 171 en 172. Hierin staan de track en sector van de plaats waar de border-files zich bevinden.

Deze plaats bestaat uit 1 sector en heeft het format van een gewone directory-pagina. Dus in byte 2 van die sector staat een soort file. In byte 5 tot 20 staat de naam en in byte 30 en 31 staat de grootte van het file. Dit repeteert om de 32 bytes, en omdat een sector 256 bytes heeft, kunnen er dus maximaal 8 files tegelijkertijd in de border staan. Het programma print alle namen, grootte en types van eventuele aanwezige betanden.

programmalisting:

```
10   open 15,8,15:open 5,8,5,"#"
11   print"0 [RVS ON]Files in kader [14
     SPATIES][RVS OFF]"
12   print#15,"b-r";5;0;18;0
13   print#15,"b-p";5;171;
14   gosub 28:tr=by:gosub28:ts=by
15   print#15,"b-r";5;0;tr;ts
16   for a=0 to
     7:print#15,"b-p";5;2+a*32;
17   gosub28:s=by:ifs=0then27
18   print#15,"b-p";5;30+a*32;
19   gosub 28:lb=by:gosub28:hb=by
20   t=t+lb+hb*256:print(lb+hb*256)
21   print#15,"b-p";5;5+a*32:forb=0to15
22   gosub 28:if by=0 then b=15:goto 26
23   printchr$(by);:next b:print,
24   id$="prg":if s and 1=1 then
     id$="USR"
25   printid$:if s and 128=128 then
     print "<";
26   print:next a:close 5,8,5:close
     15,8,15
27   print" blocks in use: ";t:end
28   get#5,a$:by=asc(a$+chr$(0)):return
```

GEOS Versie 1.3

GEOS versie 1.3, die nu al enige maanden in Nederland te koop is, verschilt vrij sterk van zijn voorganger 1.2, vooral op het gebied van de kopieerbeveiliging.

Wanneer toch wordt geprobeerd een GEOS diskette te kraken, resulteert dit vaak in de volgende problemen:

- GEOS wil geen bewerkingen meer uitvoeren op BOOT-files (GEOS, GEOS Kernal, GEOS boot)
- Wanneer geprobeerd wordt om bijvoorbeeld een file in de eerste pagina van de directory te plaatsen, dan worden alle BOOT-files gewist!
- Iedere GEOS-diskette heeft zijn eigen ID nummer (identificatie). Deze code staat verstopt en is van cruciaal belang bij het gebruik van alle GEOS applicaties, zoals GeoDEX, GeoFILE en Writers Workshop. Deze applicaties worden namelijk bij het eerste gebruik geïnitialiseerd met de ID van de gebruikte GEOS startdisk. In het vervolg zullen ze dan alleen werken als er met die startdisk is opgestart.

Deze maatregelen zijn erg fors, maar de krakers zullen al wel een antwoord hebben gevonden. Kraken en beveiligen is tenslotte een kat-en-muis-spelletje. Maar wie wordt de dupe? De argeloze koper, die de gebruikersvriendlijkheid van GEOS aanzienlijk ziet teruglopen!

Problemen met GEOS

Naar aanleiding van de grote GEOS aktie tijdens de Computer Info Beurs hebben we enige reacties ontvangen betreffende problemen met GEOS programma-diskettes.

Eén van de meest voorkomende problemen is het laden van de BOOT disk. Al gauw wordt dan door de gedupeerde gesuggereerd dat er niks op de disk zou staan of dat de inhoud incompleet is. Echter bij nader onderzoek blijkt al gauw dat het voorkomende ongemak is te wijten aan een niet goed afgestelde lees/schrijfkop van de diskdrive. Vooral de wat oudere 1541 drives leveren, na langdurig gebruik (zonder regelmatig onderhoud) lees- en schrijfproblemen op. Een ander euvel dat kan ontstaan is het ongeorganiseerd raken van de disk. Wanneer een disk door dit probleem niet meer wil inlezen, doet men er goed aan de disk d.m.v. het GEOS validate-commando te laten reorganiseren. Gebruik echter nooit het DOS validate commando, daar de disk dan inderdaad niet meer werkt. Het resultaat van het valideren is een disk, waarop alle ruimte weer beschikbaar is.

GEOS onderscheiden met Software Award

Berkeley Softworks ontving laatst de onderscheiding "Best Technical Achievement" voor hun revolutionaire disk operating system, GEOS 1.2. De onderscheiding werd toegekend door de Software Publishers Association (SPA). De SPA kan worden vergeleken met de "Academy Awards" op het gebied van de software industrie.

OPROEP!!!

Zoals u reeds heeft gemerkt, zijn we wederom gestart met een GEOS rubriek. We hopen echter, nu de opmars van GEOS in Nederland goed op gang komt, dat deze rubriek elke keer in Commodore Info kan terugkomen. Hierbij hebben we uw steun nodig. Het ligt in de bedoeling om in deze rubriek allerlei zaken betreffende GEOS aan de orde te stellen. We denken hierbij aan probleembehandeling, het geven van tips, publicatie van listings e.d. Dus als u iets te melden of te vragen heeft schrijf dan naar onderstaand adres.

Bert Venema en Peter Boncz

Voor al uw reakties:
GEOS INFO rubriek
Postbus 43048,
1009 ZA Amsterdam

Tegenwoordig wordt bijna geen enkel tijdschrift, krant of ander medium meer vervaardigd zonder tussenkomst van een computer. Steeds vaker wordt hierbij gebruik gemaakt van een DeskTop Publishing programma. Sinds kort is er voor de Commodore 64 en het GEOS besturingssysteem een DTP-pakket op de markt, GeoPUBLISH.

PERSVRIJHEID op de Commodore!

Toen Berkeley Softworks begin 1986 het grafische besturingssysteem GEOS op de markt bracht, wist men nog niet wat voor een uitbundig succes dit zou worden. De mogelijkheden die door dit nieuwe grafische besturingssysteem ontstonden, grenzen aan het onwaarschijnlijke. Al gauw volgden enkele nuttige en zeer geavanceerde applicaties, die het werken met GEOS tot een prettige bezigheid maken. Maar voor het vervaardigen van kranten, nieuwsbrieven, folders en andere grafische hoogstandjes op de Commodore 64, was men nog steeds aangewezen op pakketten als Printshop, Printmaster en Newsroom.



Terwijl de Commodore-bezitters in Nederland nog met een programma als Newsroom zaten te zweten, waren de programmeurs van Berkeley Softworks druk in de weer om hun primeur op DeskTop Publishing gebied in elkaar te sleutelen. Halverwege 1987 was het dan zover, het eerste DTP-programma voor de Commodore 64 werd geïntroduceerd GeoPUBLISH was een feit. De techniek van DTP werd al enige tijd in de PC-wereld gebruikt om tijdschriften, kranten, brochures, etc. een professioneel uiterlijk te geven. Het eerste echte DTP pakket voor de microcomputer is het nu nog zeer populaire programma Pagemaker, welke oorspronkelijk alleen op de Apple MacIntosh draaide. Tegenwoordig zijn er voor die markt zeer veel verschillende pakketten op de markt waarbij Ventura Publisher en Pagemaker wel de bekendste zijn. Het voordeel van DTP is dat men met behulp van de computer, een zgn. "ready to use" ontwerp samenstelt, compleet met logo's, tekeningen, graphics en tekst. De tekst kan natuurlijk in een of meer kolommen worden weergegeven voorzien van een fraai lettertype.



Wat is DeskTop Publishing?

Onder DTP (DeskTop Publishing) wordt verstaan: het met behulp van een computer vervaardigen van kranten, bladen of boeken. Een andere benadering van het begrip DTP is het prepareren van informatie, bestaande uit tekst en graphics (tekeningen), voor afdrukken op papier.

WYSIWYG

De term DeskTop duidt in dit geval op het feit, dat deze werkzaamheden vanuit de bureaustoel kunnen worden gedaan. Het hele concept van DTP berust op het zgn. WYSIWYG-principe, hetgeen betekent "What You See Is What You Get". Met andere woorden; u kunt het resultaat van uw werk steeds op het beeldscherm volgen. Hierbij geldt dus dat hetgeen op het scherm verschijnt, naderhand precies zo door de printer wordt afgedrukt.
Je kunt DTP ook omschrijven als een elektronische werktafel, waarop een "moederblad" wordt gemaakt, met be-

staande tekeningen en teksten. Het vervangt in feite het vel papier, de lineaal, een potlood, lijm en schaar. DTP gaat echter verder dan dat. Een DTP-programma bezit meer technische hoogstandjes, zoals het in één keer aanpassen van kopteksten in een ander lettertype of het vergroten of verkleinen van graphics. Op die manier is de gebruiker in staat zeer professionele publikaties m.b.v. een computer te verzorgen. Maar de positieve kanten van DTP houden hier nog niet op, want een gemiddeld DTP pakket is veel goedkoper dan de traditionele methodes van publiceren. Niet alleen de aanschaf is lager, maar ook de produktiekosten zijn gunstiger. Alhoewel dit laatste argument niet relevant is, aangezien het overgrote deel van de gebruikers van GeoPUBLISH zich in het hobby-circuit bevindt.
Desalniettemin, met een DeskTop Publishing programma wordt het plannen van een of meer pagina's een stuk eenvoudiger. Dit komt vooral omdat een DTP-pakket zeer flexibel is in het gebruik. Bij de traditionele manier van publiceren rijst altijd de vraag of het geschreven artikel in combinatie met plaatjes wel binnen de beschikbare ruimte past. Wanneer dit niet het geval is, resulteert dit vaak in het opnieuw typen van de tekst. Bij DTP is dit een stuk eenvoudiger, omdat er veel mogelijkheden zijn om de tekst aan de beschikbare ruimte aan te passen. GeoPUBLISH beschikt daarvoor over een ingebouwde "editor". Dit is een afgeslankte versie van GeoWRITE V2.1.

Photo Scrap

Daarnaast biedt GeoPUBLISH veel technieken; zoals het verplaatsen van blokken tekst en het wijzigen van lettertypen en hun grootte. Het gebruik van graphics (plaatjes) wordt in GeoPUBLISH helemaal kinderspel, omdat men ze op eenvoudige wijze met behulp van een "Photo Scrap" binnenhaalt en op de gewenste plaats in de pagina inpast. Naast deze mogelijkheden biedt het pakket nog meer grafische handigheidjes. Met behulp van een muis of joystick kunnen diverse tekentechnieken worden geselecteerd en toegepast. Men kan hierbij kiezen uit vierkanten, veelhoeken, cirkels en lijnen, welke men dan kan vullen met verschillende vulpatronen. Verder kan men zowel tekst als graphics op de voorgrond halen of achter een ander plaatsen. Hierbij bestaat de mogelijkheid om in een transparante modus te werken zodat alles zichtbaar blijft. Het resultaat is steeds te zien op het scherm, zodat eventuele onjuistheden meteen kunnen worden hersteld. Met GeoPUBLISH is het maken van een nieuwsbrief of iets dergelijks een plezierige bezigheid.

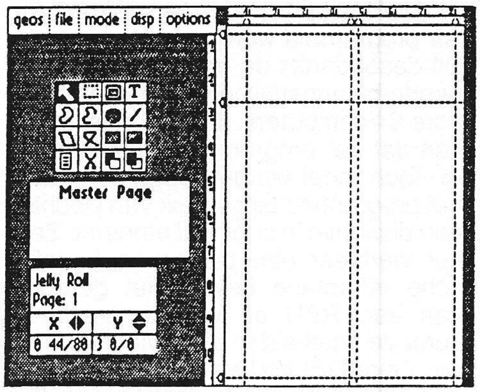


Het pakket GeoPUBLISH

Als eerste mag worden opgemerkt dat GeoPUBLISH zeer compleet wordt uitgebracht. Het vergt weinig moeite om met het pakket aan de slag te kunnen. Het GeoPUBLISH-pakket wordt geleverd in een stevige doos en bestaat uit een diskette en een Engelstalige handleiding. Dit laatste kan voor sommigen een bezwaar zijn, dat echter door het gebruiksgemak al snel wordt gecompenseerd. Voordat men begint, dient de meegeleverde diskette geïnstalleerd te worden. Het hele programma draait normaal onder Geos DeskTOP vanaf versie 1.2 of hoger. Na het installeren wordt geadviseerd een werkdiskette aan te maken en de originele diskette op te bergen.
De werkdisk bevat, afhankelijk van de gebruikte computerconfiguratie, een aantal bestanden. Echter naast een aantal optionele utilities, dienen de volgende files ten minste op de werkdiskette aanwezig te zijn: het GeoPUBLISH programma, een printerdriver, benodigde letterfonts, GeoWRITE files, een PhotoSCRAP of PhotoMANAGER en de GeoPUBLISH-libraries. Er kunnen derhalve niet te veel programma's en files op de werkdisk, want GeoPUBLISH laat, met z'n 95 Kbyte benodigde ruimte, maar weinig ruimte op diskette over. De meegeleverde handleiding beschrijft op een heldere en duidelijke wijze, hoe men diskettes in moet delen bij gebruik van een andere configuratie, zoals een tweede diskdrive of een REU (Ram Expansion Unit). De in de handleiding opgenomen tutorial neemt de aspirant uitgever aan de hand mee door de gebruiksmogelijkheden van GeoPUBLISH.

De Master-pagina

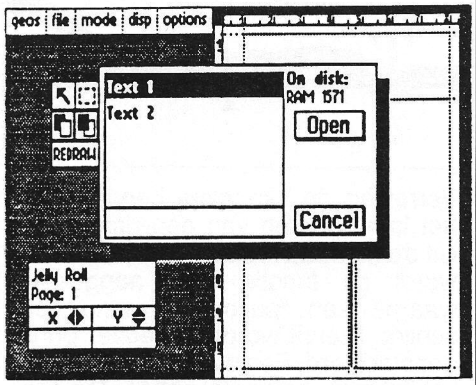
De eerste stap, nadat men een dokument heeft geopend, is het creëren van een Master-page (hoofdpagina). Deze pagina kan worden vergeleken met een soort template, die over elke af te drukken pagina wordt gelegd, alvorens deze naar de printer wordt gestuurd. In de Master-page worden zaken vastgelegd die op elke pagina weer terug moeten komen, zoals kolomindelingen, paginanummers, kop-en voetteksten en eventueel de datum. De mogelijkheden die de gebrui-



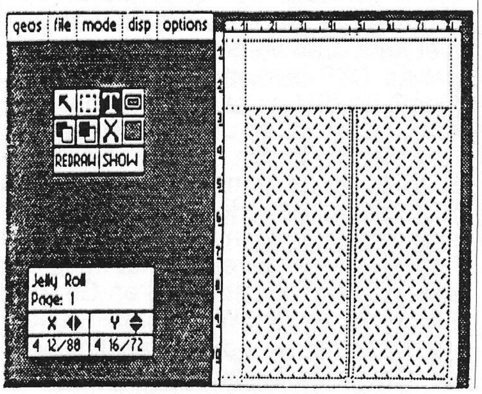
ker ter beschikking staan in de Master-page modus, zijn in een toolbox weergegeven en kunnen door het aanklikken met de muis-pointer worden geactiveerd.
Men selecteert gebieden op de Master-page om deze later te vullen met plaatjes. Om een symmetrisch geheel te krijgen, is de mogelijkheid aanwezig om "guidelines" (hulplijnen) in te stellen. Helaas werkt GeoPUBLISH voor wat betreft de indeling van een pagina nog steeds in inches. Op zich is dit echter niet storend. Zeker niet wanneer men op duidelijke wijze steeds kan aflezen waar de pointer zich bevindt, d.m.v. het dokument indicatie venster. Als men eenmaal een gebied heeft aangegeven, kan worden gekozen voor tekst of graphics. Wanneer tekst ingevoerd moet worden, klikt men op het icoon met de hoofdletter T. Het graphics symbool lijkt veel op het icoon van een GeoPAINT-file. Afhankelijk van uw wensen kunt u een ontworpen Master-page opslaan in een library (bibliotheek). Dit is vooral handig als u bijvoorbeeld een clubblad moet maken en steeds dezelfde pagina-indeling en lay-out wilt gebruiken. GeoPUBLISH wordt standaard geleverd met 42 layouts, waarvan 21 voor Master-pagina's en 21 voor Page-layouts geschikt zijn. De functie Master-page Layouts biedt de mogelijkheid om allerlei tekenwerk te verrichten in de zojuist ontworpen Master-pagina. Dit kan bijvoorbeeld een scheidingslijn zijn tussen twee kolommen, of een koptekst. Tekeningen en graphics die op elke pagina terug moeten keren, worden nu ook gemaakt. Men denkt hierbij bv. aan logo's

Pagina-opmaak

Er zijn twee manieren waarop men de uiteindelijke pagina-opmaak kan realiseren. De eerste is de Page-Layout, waarbij tekst en plaatjes worden geïmporteerd uit GeoWRITE-files of een Photo Scrap. De Page-Layout modus biedt geen tekenmogelijkheden, maar wel diverse opmaakfaciliteiten. Een handig hulpmiddel in deze modus is de "snap" functie. Deze functie zorgt ervoor dat een te definiëren gebied gelijkgesteld wordt met de "guidelines" (hulplijnen). Hierdoor worden de naderhand in te voeren kolommen tekst allemaal van gelijke grootte. Hebben we eenmaal een gebied gemarkeerd, dan kunnen we deze vullen met tekst uit een GeoWRITE file. Hiertoe klikt men eerst op het pijl-icoon om het gebied te activeren. Vervolgens klikt men met de "pointer" het T-icoon aan, waarna een window met de beschikbare files verschijnt.



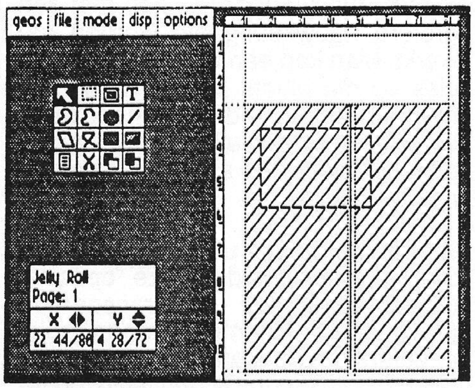
Men selecteert het benodigde file en de tekst zal als een gearceerd veld in de geselecteerde kolom verschijnen. Om de tekst over te laten vloeien naar de volgende kolom, klikt men eenvoudig binnen de omlijning van de volgende kolom. Ook hier wordt d.m.v. een arcering aangegeven dat de kolom is gevuld met tekst. Voordat de tekst bekeken kan worden, dient deze eerst "geripppled" te worden. Hieronder wordt verstaan, het aanpassen van de tekst aan het GeoPUBLISH formaat. Dit betekent dat de tekst wordt geformeerd om eventueel ingevoegde plaatjes e.d. Een ononderbroken arcering geeft aan dat de tekst is "geripppled". Het resultaat kan alleen worden bekeken wanneer het dokument zich in de Page-Graphics modus bevindt.



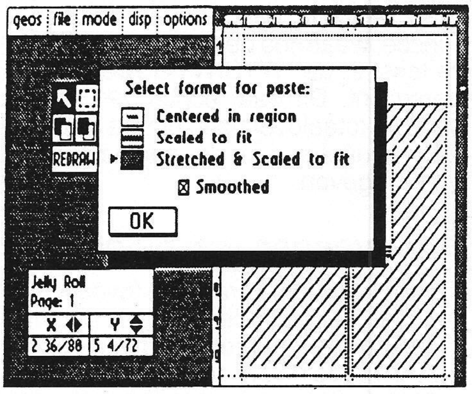
Men beschikt dan over de "zoom"-functie, waarmee de nu nog onleesbare tekst in de "WYSIWYG"-weergave verschijnt. Dit gaat echter ten koste van het totaaloverzicht, omdat slechts een gedeelte van de pagina wordt weergegeven.

Het invoegen van plaatjes.

Terug naar de "preview"-(overzicht) functie, waarbij de gehele pagina weer op het scherm wordt getoond. Het invoegen van een "bit mapped" graphic (tekening opgebouwd uit puntjes) kan op zeer eenvoudige wijze worden gerealiseerd. Men selecteert wederom een gebied ergens tussen bijvoorbeeld twee kolommen. Vervolgens klikt men het "redraw" icoon aan, waardoor de tekst opnieuw wordt weergegeven als een onderbroken gearceerd gebied. Hierdoor kan de tekst zich aanpassen aan de omvang van het in te voegen plaatje. Nu halen we het plaatje binnen door eerst het gebied te markeren en vervolgens het "graphic" icoon aan te klikken.
Het programma toont nu een window op het scherm, waaruit een keuze geselecteerd kan worden op welke wijze het plaatje moet worden ingevoegd. Deze mogelijkheden zijn gecentreerd in het gebied, aangepast aan de beschikbare ruimte en op een wijze zodat het plaatje wordt uitgerekt en aangepast aan de beschikbare ruimte.



Hierbij bestaat tevens de mogelijkheid om dit "smoothed" (geleidelijk) te laten gebeuren. Met het "text" icoon kunnen we nog een "headline" (artikelkoptekst) invoeren, die rechtstreeks vanaf het toetsenbord d.m.v. een editor wordt ingetypt. Een scala aan variëteiten staat de gebruiker hierbij ter hand. Zo kan men de letterhoogten laten variëren van 1 tot 192 points, een keuze omtrent een lettertype is aanwezig, de weergave van het gekozen lettertype is een van de opties.
Niet te vergeten, is de nuttige optie, waarbij de in te typen tekst horizontaal of verticaal, alsmede van onder naar

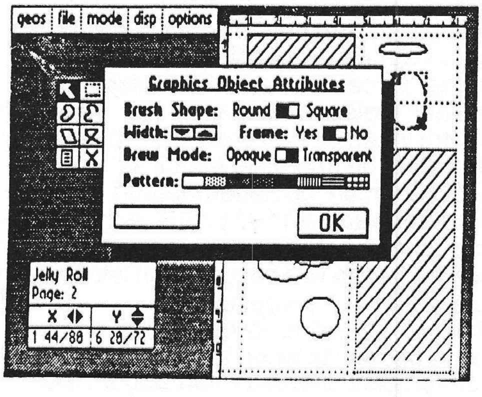


boven of vice versa kan worden weergegeven. Ook het patroon van de tekst waarmee de gebruiker geavanceerde schaduwteksten kan vervaardigen kan worden geselecteerd. Op dezelfde wijze worden vervolgpagina's gemaakt.

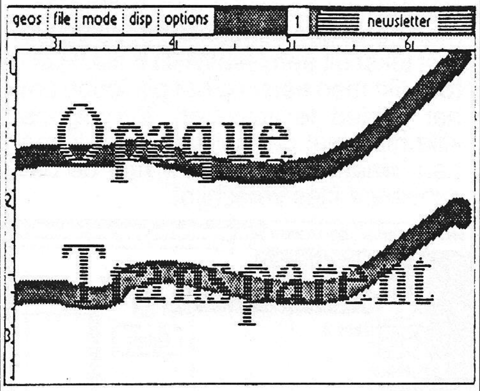
Cirkels, lijnen, enz.

Nog steeds in de Page-Graphic modus wordt een pagina voorzien van versieringen, zoals cirkels, ellipsen, vierkanten e.d. Deze mogelijkheden worden geselecteerd uit de "toolbox" en blijven actief tot een andere functie wordt aangeklikt. Met het "attribute" icoon kan het vulpatroon gekozen worden om de tekening te maken of in te kleuren. En dan tekent men het gewenste figuur in een van tevoren geselecteerd gebied. Wanneer men de "zoom" functie gebruikt, kan met een grotere nauwkeurigheid worden gewerkt. Men kan een tekening dan precies op die plaats zetten, waar men hem wil hebben. Indien het noodzakelijk is kan men een afbeelding nog verplaatsen naar een andere plaats binnen de pagina.

Een aardige optie uit het "graphic attribute" window is de keuze "opaque" (ondoorschijnend) of "transparant" (doorzichtig). Hiermee kunnen leuke effecten in het tekenwerk worden bereikt. Er zijn natuurlijk nog legio mogelijkheden die dit DTP-pakket voor de Commodore 64 tot een zeer uitzonderlijk programma maken. Het zou te ver voeren deze mogelijkheden in dit artikel allemaal te behandelen.



Afdrukken van een dokument.

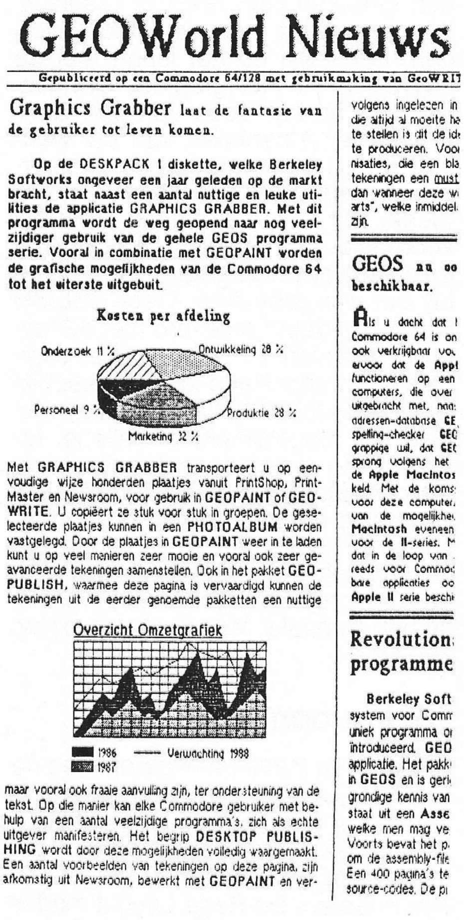


Verreweg de simpelste handeling is het laten printen van een dokument. Uit de menubalk, bovenin het scherm, wordt de functie "file" aangeklikt, waarna een "pulldown" menu verschijnt. Hieruit wordt de keuze "print" geselecteerd. Een dialoogvenster verschijnt met een aantal opties. Ten eerste kan gekozen worden van welke tot welke pagina afgedrukt moet worden. Voorts kan men kiezen of afgedrukt moet worden op losse vellen of kettingpapier. Naast deze opties kan een selectie worden gemaakt om alleen de Master, de layout of de gehele inhoud af te drukken. Op een matrixprinter van 9 x 11 dots is het resultaat toch zeer redelijk te noemen. Met een laserprinter, waarvoor een speciale printer-driver wordt meegeleverd, is het uiteindelijke produkt gelijk aan een afdruk gemaakt met behulp van een PC of een Apple MacIntosh.

Conclusie.

GeoPUBLISH is, en hoe kan het ook anders, een goed doordacht programma. Berkeley Softworks is er wederom in geslaagd een uniek pakket op de markt te brengen, dat z'n weerga niet kent. Een vergelijking met soortgelijke DTP-pakketten leert ons dat GeoPUBLISH met kop en schouders boven de anderen uitsteekt. Het aantal mogelijkheden welke GeoPUBLISH kent, zou men alleen op een PC verwachten. Het aantal toepassingsmogelijkheden met GeoPUBLISH zijn onbegrensd. Het integreren van files uit GeoCALC en GeoFILE (zij het via GeoWRITE) beïnvloedt de bruikbaarheid op een gunstige wijze. GeoPUBLISH zal, naast de andere applicaties van GEOS, een populair programma worden in Nederland. Dit desondanks de enigszins teruglopende belangstelling van de Commodore 64 computers. Je zou haast denken dat het programma zaligmakend is. Toch moet worden opgemerkt dat het programma bij gebruik van slechts een diskdrive in snelheid afneemt. Zeker wanneer een pagina veel grafische informatie bevat. Het gebruik van een REU of tweede diskdrive komt de snelheid in dit geval ten goede. GeoPUBLISH vormt in elk geval een stap in de goede richting, van de verdere volwassen wording van de Commodore 64 en 128. Met dit pakket kan de tot nog toe amateuristische "uitgever" werken aan een kwalitatief hoogwaardig eindresultaat. Weliswaar binnen de grenzen van de mogelijkheden van de C64. En niet in de laatste plaats, de creativiteit en fantasie die de gebruiker er zelf in legt. GeoPUBLISH wordt geïmporteerd door FROTECH BV. (Postbus 189, 3620 AD Breukelen). Het pakket komt medio juni in Nederland op de markt en gaat *f* 159,00 kosten, hetgeen voor een DTP-pakket van deze allure zeker niet duur is.

Bert Venema



GEOWorld Nieuws

Gepubliceerd op een Commodore 64/128 met gebruikmaking van GeoWRIT

Graphics Grabber laat de fantasie van de gebruiker tot leven komen.

Op de DESKPACK 1 diskette, welke Berkeley Softworks ongeveer een jaar geleden op de markt bracht, staat naast een aantal nuttige en leuke utilities de applicatie GRAPHICS GRABBER. Met dit programma wordt de weg geopend naar nog veelzijdiger gebruik van de gehele GEOS programma serie. Vooral in combinatie met GEOPAINT worden de grafische mogelijkheden van de Commodore 64 tot het uiterste uitgebuit.

Kosten per afdeling

Onderzoek 11 %
Ontwikkeling 28 %
Personeel 9 %
Produktie 28 %
Marketing 32 %

Met **GRAPHICS GRABBER** transporteert u op eenvoudige wijze honderden plaatjes vanuit PrintShop, PrintMaster en Newsroom, voor gebruik in **GEOPAINT** of **GEOWRITE**. U copiëert ze stuk voor stuk in groepen. De geselecteerde plaatjes kunnen in een **PHOTOALBUM** worden vastgelegd. Door de plaatjes in **GEOPAINT** weer in te laden kunt u op veel manieren zeer mooie en vooral ook zeer geavanceerde tekeningen samenstellen. Ook in het pakket **GEOPUBLISH**, waarmee deze pagina is vervaardigd kunnen de tekeningen uit de eerder genoemde pakketten een nuttige

Overzicht Omzetgrafiek

1986
1987
Verwachting 1988

maar vooral ook fraaie aanvulling zijn, ter ondersteuning van de tekst. Op die manier kan elke Commodore gebruiker met behulp van een aantal veelzijdige programma's, zich als echte uitgever manifesteren. Het begrip **DESKTOP PUBLISHING** wordt door deze mogelijkheden volledig waargemaakt. Een aantal voorbeelden van tekeningen op deze pagina, zijn afkomstig uit Newsroom, bewerkt met **GEOPAINT** en ver-

volgens ingelezen in
die altijd al moeite ha
te stellen is dit de id
te produceren. Voo
nisaties, die een bla
tekeningen een must
dan wanneer deze w
arts", welke inmiddel
zijn.

GEOS nu oo
beschikbaar.

Als u dacht dat
Commodore 64 is on
ook verkrijgbaar vo
ervoor dat de Appl
functioneren op een
computers, die over
uitgebracht met, naa
adressen-database GE
spelling-checker GEO
grappige wil, dat GEO
sprong volgens het
de Apple MacIntos
keld. Met de koms
voor deze computer
van de mogelijkhe
Macintosh eveneen
voor de II-series. M
dat in de loop van
reeds voor Commod
bare applicaties oo
Apple II serie besch

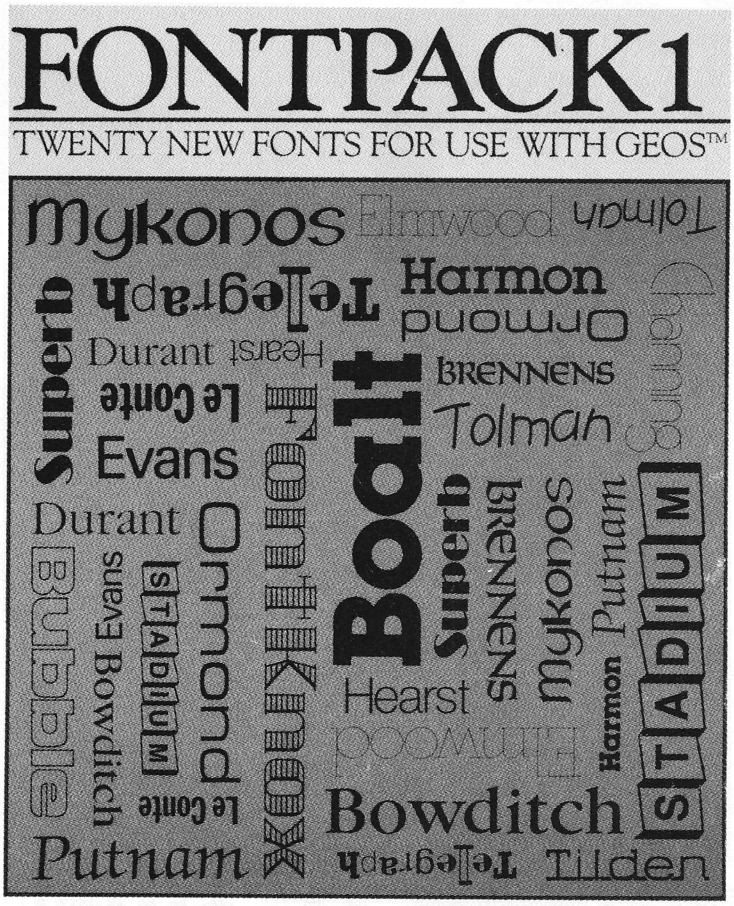
Revolution
programme

Berkeley Soft
system voor Comm
uniek programma o
introduceerd. GEO
applicatie. Het pakk
in GEOS en is geri
grondige kennis van
staat uit een Asse
welke men mag ve
Voorts bevat het p
om de assembly-file
Een 400 pagina's te
source-codes. De p

Deze nieuwe aflevering uit een reeks technische artikelen over GEOS door Peter Boncz, gaat over fonts. Er wordt uitgebreid ingegaan op de bouw en de FILE-structuur van deze fonts. Ook het VLIR systeem (Variable Length Indexed Records) komt daarbij weer aan de orde. De listing van deze keer is een FONT-MAKER, waardoor het op simpele wijze mogelijk wordt zelf nieuwe lettertypes te ontwerpen en te gebruiken.

Zelf GEOS-fonts ontwerpen

In Geowrite en Geopaint heeft men de beschikking over een aantal verschillende fonts, ofwel lettertypes. Deze fonts zijn meestal in meerdere groottes aanwezig, en kunnen in verschillende stijlen op het scherm en de printer gebracht worden. BSW (Berkeley SoftWorks) is het enige font dat standaard is, het is 'memory resident', oftewel altijd in het geheugen aanwezig. Het behoort tot de GEOS Kernal, en wordt voor Dialog Boxes en Pull-down menus in de GEOS- environment van DESKTOP gebruikt.



De andere fonts zijn als aparte files op disk aanwezig, zij kunnen - als de gebruiker een beroep op ze doet - in het geheugen 'ge-download' worden. Normaal zijn er 5 van zulke fonts, iedereen die wel eens met GEOS gewerkt heeft kent ze: University, Roma, California, Dwinelle en Cory.

Omdat variatie nooit weg is heeft Berkeley Softworks een zogenaamd Fontpack uitgebracht met 20 extra lettertypes. Degenen die deze tekst goed lezen en eventueel de listing intikken, kunnen daar echter nog ontelbare aan toevoegen. In dit artikel wordt namelijk het precieze hoe en waarom van fonts in GEOS uitgelegd. De listing van deze keer is bovendien een zg. font-maker. Maar wat is er nu bijzonder aan GEOS-fonts? Het eerste wat opvalt is dat lettertypes in bijvoorbeeld Geowrite werken met proportional spacing. Dit houdt onder andere in dat een 'm' breder is als een 'i'. Voorts is het zo dat de computer uit het ingeladen font de verschillende styles (bijvoorbeeld BOLD of ITALIC) berekent, maar dat er geen sprake is van een vergroot- of verklein-routine. Voor iedere puntgrootte in een font wordt dus een aparte karakterset gegeven.

grootte

De grootte van een karakter in GEOS wordt aangeduid in puntjes. Een puntje is op het scherm 1/72 inch, en op een normale 80 koloms printer 1/80 inch. Deze grootte slaat op de hoogte. De breedte van een karakter is, zoals gezegd, variabel. Verder van belang is nog de ondermarge. Bij normale karakters wordt een ondermarge aangehouden, d.w.z. er is een lege ruimte aan de onderkant van een karakter. Dit met het doel letters als de 'p' de mogelijkheid te geven naar onderen uit te steken. De hoogte van een karakter in GEOS is niet aan grenzen gebonden, de maximale breedte is daarentegen 54 punten.

karakterset

Een reeks karakters in dezelfde stijl wordt een karakterset genoemd. Een font is een verzameling karaktersets van dezelfde stijl maar met verschillende groottes. Een karakterset wordt daarom naast zijn naam aangeduid met de puntgrootte, bijvoorbeeld Roma 9. Ieder font heeft een eigen font identifikatie-nummer. Fontnummers zijn 10 bits lang. Legale fontnummers varieren van 0 tot 1027. Iedere karakterset heeft zijn eigen font-ID. Een font-ID is een combinatie van het fontnummer en de puntgrootte van de karakterset. Bits 15 tot 6 bepalen het fontnummer, bits 5 tot 0 bepalen de puntgrootte. In het schema van figuur 1 staan de nummers, groottes en font ID's van de karaktersets in de fonts die standaard bij GEOS geleverd worden. In GEOS bestaat een karakterset normaliter uit 96 tekens,

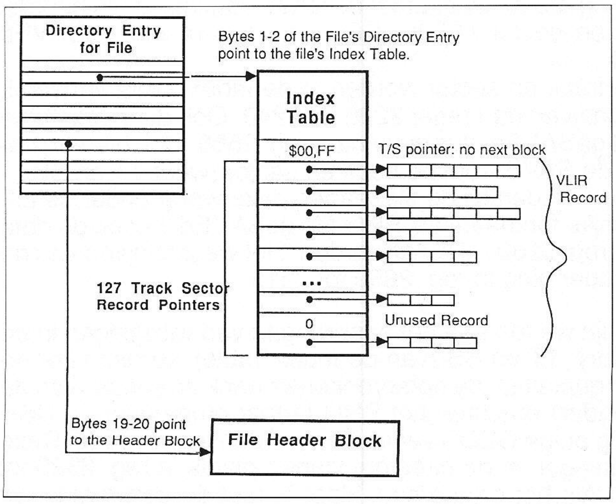
die aangeduid worden met de waarden 32 tot 127. GEOS codes 64 tot 95 komen overeen met ASCII 128 tot 159, de codes 96 tot 127 met ASCII 64 tot 95. De waardes 32 tot 63 zijn in GEOS en ASCII identiek van betekenis. De andere codes, vooral de lage nummers, worden in Geowrite gebruikt voor het aanroepen van functies, of het aangeven van bijv. tabstops, verandering in font of stijl, en nieuwe pagina markeringen.

GEOS Fonts			
Font Name	Number Top 10 bits	Point Sizes Last 6 Bits	ID Word
BSW	0	9	$00 09
University	1	6 10 12 14 18 24	$00 46 $00 4A $00 4C $00 4E $00 52 $00 58
California	2	10 12 13 14 18	$00 8A $00 8C $00 8D $00 8E $00 92
Roma	3	9 12 18 24	$00 C9 $00 CC $00 D2 $00 D6
Dwinelle	4	18	$01 12
Cory	5	12 13	$01 4C $01 4D

1. Overzicht van de GEOS-fonts

VLIR

Fonts worden op disk opgeslagen in VLIR files van het type FONT. Iedere VLIR file, dit kwam in de vorige aflevering al aan de orde, bestaat uit 3 delen, nl. de Fileheader, de Index Table en de afzonderlijke Records. In de directory wordt achter de filenaam verwezen naar track en sector van de fileheader. De 2 bytes voor de filenaam verwijzen naar de Index Table. Iedere GEOS file is opgedeeld in RECORDS, deze komen in deze Index Table samen. Het VLIR-structuur schema (ill. 2) werd vorige keer ook geplaatst, maar is van groot belang zodat het er deze keer weer bijstaat.



2. GEOS-VLIR opslagsysteem

Fileheader

Informatie die nodig is om een font te gebruiken staat in de fileheader. Zoals gebruikelijk bevat deze de data van het icon (byte 5 tot 67), en van bijv. het laad-adres en start-adres. Specifiek voor een font zijn echter de bytes 97 tot 160 (vanaf 0 gerekend: de eerste byte is dus byte 0). Bytes 97 tot 127 herbergen de fontgrootte-tabel. Daarin staat de ruimte in bytes die de karaktersets (van klein naar groot) in het geheugen innemen. De bytes 128 en 129 bevatten het fontnummer, en tot slot bevindt zich de font ID-tabel in de bytes 130 tot 160. In deze tabel staat het font ID van iedere karakterset van de kleinste naar de grootste puntgrootte. Eventuele lege plaatsen in deze tabellen worden opgevuld door $00's (zie ill. 3).

Offset into File Header	# Bytes	Description
OFF_GHFONTID = 128 ($80)	2	Unique Font ID
OFF_GHPOINT_SIZES = 130 ($82)	32	Font ID for each character set, from smallest to largest. Padded with 0s.
OFF_GHSET_LENGTHS = 97 ($61)	32	Size in bytes of each character set from smallest to largest. Padded with 0s.

3. fontdata in de fileheader.

Record-structuur

De Index Table beslaat 1 block van de disk (256 bytes). De eerste 2 bytes bevatten de waardes $00 en $ff. Track en sector van Record 0 staan in de derde en vierde byte, die van Record 1 in byte 5 en 6 etc. Iedere karakterset is een record van de VLIR file. De karakterset met een grootte van n puntjes staat in het n-de record van de file, dus record 15 bevat de karakterset met puntgrootte 15. Als er meerdere lettergroottes beschikbaar zijn, dan bestaan er dus ook meerdere records, ieder record bevat dus 1 karakterset.

Zo'n record kan in drie delen worden gedeeld:

- byte 2 tot 9 huisvesten enige variabelen, zie ill. 4. (In byte 0 en 1 staan altijd de track en sector waar het record verdergaat.)
- in byte 10 tot 203 staat de width-table, waarin de breedte van de afzondelijke letters wordt vastgelegd.
- De rest van de bytes bevatten het data-blok, daat worden de karakters puntje voor puntje gedefinieerd.

Data-blok

Om het allemaal duidelijk te houden is het hier beter even bij het laatste onderdeel van een record, het data-blok, te beginnen. Een font wordt door GEOS per horizontale rij puntjes weergegeven. Als de puntgrootte (hoogte) 8 is zijn er dus acht horizontale rijen data. Zo'n horizontale rij bestaat dus uit een aantal bits dat gelijk is aan alle afzonderlijke letterbreedtes bij elkaar opgeteld. Het aantal bytes dat zo'n horizontale rij inneemt, is dit aantal bits gedeeld door acht, afgerond naar boven. Dit aantal staat ook bij de variabelen, in byte 3 en 4. De geheugenruimte, die de karakterset inneemt is logischerwijze horizontaal maal vertikaal, dus het aantal bytes per horizontale strook maal de puntgrootte.

Width-table

De width-table staat middenin het record: achter de variabelen (gewoonlijk dus in byte 10, zie schema 4), en voor het data-blok. Deze tabel bestaat (als er 96 tekens zijn) uit 97 breedte- pointers. De width-table werkt heel eenvoudig. De eerste breedte- pointer is $00, $00. Als karakter 32 (het eerste teken) bijv. 8 puntjes breed is, is de volgende breedte-pointer $08, $00. Dit gaat volgens het normale adresseringssysteem, dus als de low-byte over de 255 komt wordt de high-byte 1 opgehoogd. De eerste breedtepointer geeft aan waar het eerste karakter begint etc, dus de 96e waar het 96e karakter begint en de 97e breed-

disk-struktuur van een record

byte	functie/inhoud
$02	puntgrootte min ondermarge min 1
$03+$04	aantal bytes per horizontale strook
$05	puntgrootte
$06+$07	pointer (low-byte, high-byte) naar begin width-table, gewoonlijk $08, $00.
$08+$09	pointer naar begin data-blok, gewoonlijk $CA, $00. (de pointers wijzen vanaf byte $02).
$0A+$0B	eerste breedte-pointer.
etc...	
$CA+$CB	laatste (97e) breedte-pointer.
$CC	begin data-blok

4. record-structuur, data-blok en width-table

te-pointer geeft aan waar het 96e karakter eindigt. Direct na de width-table begint dan het data-gedeelte.

Samenvatting

Dit was het dan weer voor deze keer wat de droge informatie betreft. Fonts en VLIR in GEOS blijven taaie materie, daarom volgt hier een beknopte samenvatting:

- een font is een verzameling karaktersets met verschillende hoogten.
- een font wordt op VLIR manier op disk gezet, dus er is een FILEHEADER en een INDEX TABLE waar de RECORDS samenkomen.
- iedere karakterset is een record, een karakterset van een n-punts letter is het n-de record van de index table.
- een record is op te delen in: variabelen, width-table, en een datablok.
- fontdata worden per horizontale rij bytes opgeslagen.

Praktijk

De theorie is beproken en (hopelijk) duidelijk, nu dan naar de praktijk! De praktijk houdt traditiegetrouw een listing in, en dit keer is dat een hele mooie utility: een fontmaker. Wat kan hij wel, en wat kan hij niet? Het programma is in BASIC geschreven en is weliswaar vrij overzichtelijk, maar het werkt dus niet al te snel. Bij het ontwerpen van grote lettertypes (bijv. 20 x 20 puntjes), kan het bovendien gebeuren dat de BASIC-ruimte voor variabelen (33 K!) vol raakt. Dit gebeurt echter alleen als alle 96 tekens een dergelijke breedte krijgen, dat is niet echt waarschijnlijk. Toch is het aan te raden regelmatig het font-in- aanbouw te saven, want je weet natuurlijk nooit. Een andere beperking van deze fontmaker is het feit, dat hij alleen fonts kan maken met 1 lettergrootte. Als echter ook in de mogelijkheid van meerdere puntgroottes was voorzien, zou de listing echt te omvangrijk geworden zijn. Maar de FONT-MAKER is toch een vrij comfortabel programma, dat dingen kan, die tot nu toe niet mogelijk waren. Bij mijn weten bestaat er ook geen literatuur over de structuur van GEOS-fonts, dus Commodore Info heeft de première. Al met al een heel aardige listing, er kan bijvoorbeeld een FONTPACK van *f* 75,- mee uitspaard worden. Tot een volgende keer!

Peter Boncz

De FONT-MAKER

regeloverzicht font-maker:

reg. 1000 tot 1710 - het EDIT gedeelte
reg. 1720 tot 1840 - HOOFDMENU
reg. 1850 tot 1970 - LADEN en SAVEN
reg. 1980 tot 2740 - CONVERSIE
reg. 2750 tot 2840 - DATA

Omdat het ontwerpen van een font een heel karwei is, is het EDIT- gedeelte vrij comfortabel uitgevoerd; er zijn weinig dingen die men fout kan doen. Bovendien is er de mogelijkheid tussentijds te saven, zodat het ontwerpen van een font niet in een ruk moet gebeuren. Als het programma gestart wordt komt men in het hoofdmenu terecht. Kiest men dan editen, dan volgt de vraag naar de maximale hoogte, breedte en de ondermarge. Denk er wel om tijdens het editen die marge aan te houden, en houdt er ook rekening mee dat die marge inbegrepen is bij de hoogte. Dus een hoogte van 10 en een marge van 2 levert een letter van 8 puntjes op.

Tijdens het editen hebben de volgende toetsen functies:
F1 - naar het volgende ASCII teken.
F3 - naar het vorige ASCII teken.
F5 - naar het HOOFDMENU.
F7 - letterbreedte 1 punt smaller.
F8 - letterbreedte 1 punt breder.
RETURN - veranderen van vol in leeg of omgekeerd.
De cursortoetsen besturen de cursor.

Programma-uitleg

In het EDIT-gedeelte worden de karakters opgeslagen in de array A$ [asc,ho], waarin asc het ASCII teken is, en ho de hoogte. Deze array bestaat uit puntjes en sterretjes (een puntje betekent 'leeg', een sterretje 'vol'). De lengte van die array-string (ofwel de breedte van het teken) is per karakter verschillend en wordt opgeslagen in de array br [asc].

Het CONVERSIE-gedeelte werkt deze gegevens om. In reg. 2130 tot 2180 worden de array-strings A$[asc,ho] van dezelfde ho(ogte) achter elkaar gezet van asc = 32 tot asc = 127. Ze worden omgewerkt naar een serie bytes die in het geheugen gePOKEd wordt (Het geheugengebied waarin gePOKEd wordt is adres $C000 tot $D000, zet daar dus geen snelladers of disk-monitoren neer!). Zo krijg je een n-aantal (n=hoogte) series bytes. Daarvoor, in reg. 2060 tot 2090, waren al de 8 variabelen ingePOKEd, en in reg. 2100 tot 2120 waren de breedtes (opgeslagen in br[asc]), omgewerkt tot de width- table. Omdat de variabelen, de width-table en het data-blok dan klaar zijn, wordt een begin en eindadres aangegeven (BB en EE) en gesprongen naar reg. 2550. Daar staat de SAVE-routine. Deze bepaalt track en sector (TT en SS) waar het record geSAVEd wordt.

Deze track en sector worden in de Index Table verwerkt, we schrijven dan regel 2200 tot 2240. Ook de Index Table wordt geSAVEd, dus weer naar reg. 2550. Ook hiervan bepaalt de SAVE- routine track en sector (weer TT en SS). We komen dan bij de fileheader, deze wordt onder de uiteindelijke fontnaam als PRG-file geSAVEd. Let op de data die in reg. 2260 reREAD worden. Het wegschrijven van de fileheader ging in reg. 2250 tot 2310.

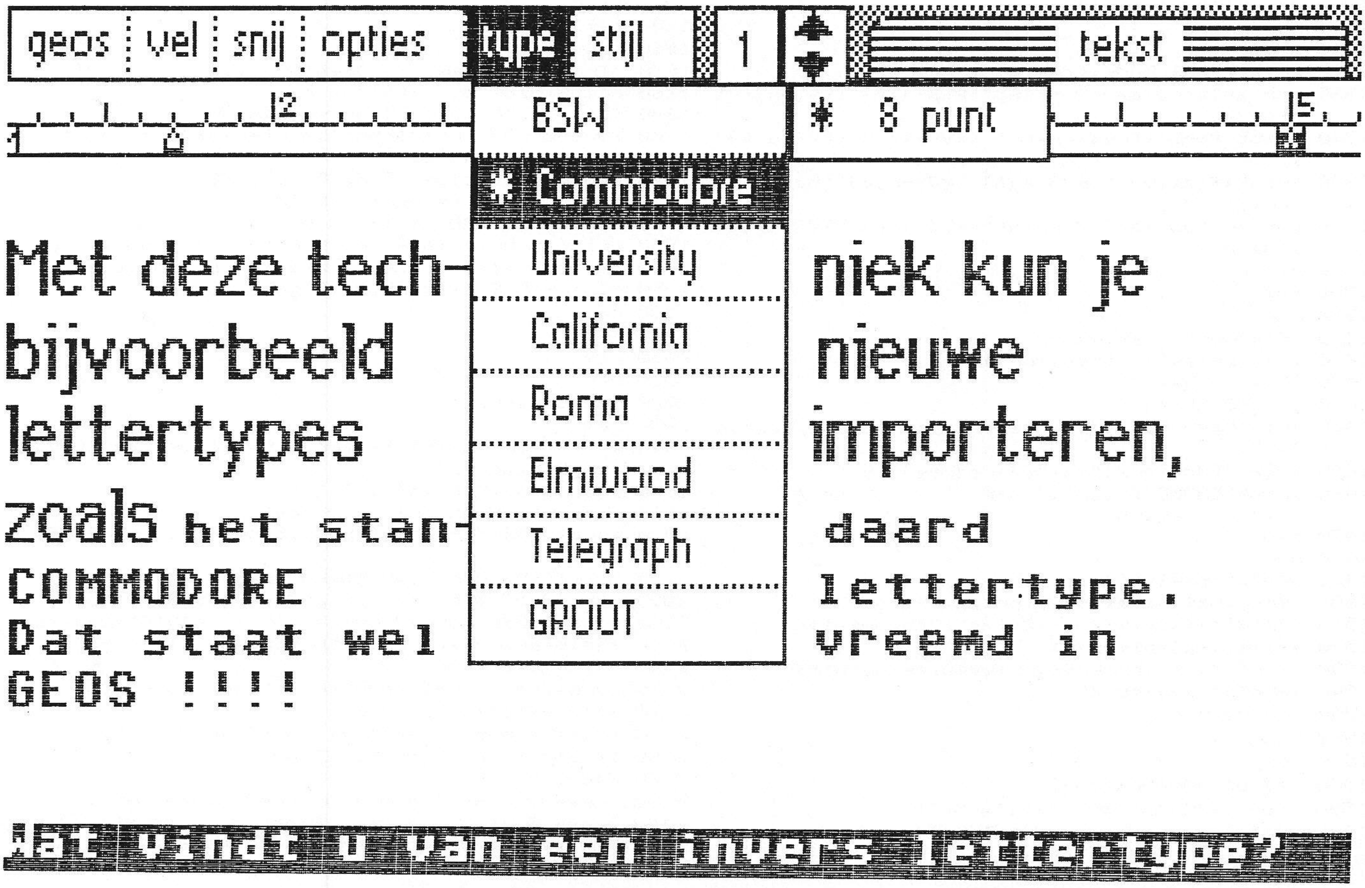
Dan zijn we toe aan het aanbrengen van wijzigingen in de directory. TT en SS (van de Index Table) worden voor de naam geplaatst, de oorspronkelijke track en sector (van de fileheader) erachter, het PRG-format omgetovert in USR en nog enige GEOS variabelen worden toegevoegd. Deze omzettingen in de directory vinden plaats in reg. 2320 tot 2460. Als het programma klaar is met de conversie, kan GEOS opgestart worden.

Listing Fontmaker

```
1000 rem
1010 rem
1020 rem
1030 rem
1040 rem invoergedeelte
1050 rem
1060 goto1780
1070 input"max. hoogte,breedte";ho,br
1080 if ho<0orho>22orho<>int(ho)orbr<>int(br)
     orbr<0orbr>35then1070
1090 input"ondermarge";om
1100 gosub2730:fora=0to95:br(a)=1
1110 forb=1toho:a$(a,b)=".":nextb:nexta
1120 xx=br:yy=ho:gosub1320
1130 x=3:y=3:gosub1400
1140 lb=983+x+y*40:pokelb,peek(lb)+128
1150 geta$:ifa$=""then1150
1160 lb=983+x+y*40:pokelb,peek(lb)-128
1170 i=1:ifa$="[neer]"theny=y+1:i=0
1180 ifa$="[op]"theny=y-1:i=0
1190 ifa$="[rechts]"thenx=x+1:i=0
1200 ifa$="[links]"thenx=x-1:i=0
1210 ifa$="[ F1 ]"thenas=as+1:i=2
1220 ifa$="[ F7 ]"thengosub1580
1230 ifa$="[ F8 ]"thengosub1620
1240 ifa$="[ F3 ]"thenas=as-1:i=2
1250 ifa$="[ F5 ]"oras<0oras>95thengoto1780
1260 if a$=chr$(13)thengosub1490
1270 if i>0thengosub1320
1280 ifi=2thengosub1680
1290 gosub1400:goto1140
1300 rem
1310 rem
1320 print"[CLR-HOME][neer] ";
1330 for a=1tobr(as)+2:print"[RVS-aan] [RVS-uit]"
     ;:next
1340 print:forb=1toyy:print" [RVS-aan] [RVS-uit]"
     ;
1350 printa$(as,b);"[RVS-aan] [RVS-uit]":next
     :print" ";
1360 for a=1tobr(as)+2:print"[RVS-aan] [RVS-uit]"
     ;:next
1370 return
1380 rem
1390 rem
1400 if(y>ho+2)theny=ho+2
1410 if(x>br(as)+2)thenx=br(as)+2
1420 if(x<3)thenx=3
1430 if(y<3)theny=3
1440 print"[HOME][12xrechts][2xspatie]y:[5xspatie
     ]x:";
1450 print"[8xlinks]";y;"[7xrechts]";x
1460 print"[HOME][rechts]char :";as+32;chr$(p
     p);" ":return
1470 rem
1480 rem
1490 d$=".":b$=left$(a$(as,y-2),x-3)
1500 c$=right$(a$(as,y-2),br(as)-x+2)
1510 ifmid$(a$(as,y-2),x-2,1)="."thend$="*"
1520 a$(as,y-2)=b$+d$+c$
1530 if d$="."thenpoke983+x+y*40,46:goto1550
1540 poke983+x+y*40,42
1550 i=0:return
1560 rem
1570 rem
1580 x=3:br(as)=br(as)-1
1590 ifbr(as)<1thenbr(as)=1:goto1670
1600 fora=1toho:a$(as,a)=left$(a$(as,a),br(as
     ))
1610 next:return
1620 br(as)=br(as)+1
1630 ifbr(as)>brthenbr(as)=br:goto1650
1640 fora=1toho:a$(as,a)=a$(as,a)+".":next
1650 return
1660 rem
1670 rem
1680 if as>32and as<58thenpp=as+64:goto1710
1690 if as>64and as<91thenpp=as:goto1710
1700 pp=as+32:ifpp=34thenpp=39
1710 return
1720 rem
1730 rem
1740 rem
1750 rem
1760 rem hoofdmenu
1770 rem
1780 print"[CLR-HOME][2xneer][RVS-aan][15xspatie]H
     oofdmenu[16xspatie][RVS-uit]"
1790 print"[2xneer][7xspatie]1) editen":print"[neer]
     [7xspatie]2) saven":print"[neer][7xspatie]3)
     laden"
1800 print"[neer][7xspatie]4) conversie":print"[2xne
     ][RVS-aan][13xspatie]Maak uw keuze[14xspatie]
     [RVS-uit]"
1810 geta$:a=val(a$):ifa$=""then1810
1820 as=0:ifa=1andho<>0then1120
1830 on a goto 1070,1930,1870,2040,1780
1840 goto 1780
1850 rem
1860 rem
1870 clr:input"naam";f$:open15,8,15
1880 open8,8,8,"@0:"+f$+",s,r"
1890 input#8,br:input#8,ho:gosub2730
1900 forb=0to95:fora=1toho:input#8,a$(b,a):next
1910 input#8,br(b):next
1920 close8,8,8:close15,8,15:goto1780
1930 input"naam";f$:open15,8,15
1940 open8,8,8,"@0:"+f$+",s,w"
1950 print#8,br:print#8,ho:forb=0to95:fora=1tono
1960 print#8,a$(b,a):nexta:print#8,br(b):nexto
1970 close8,8,8:close15,8,15:goto1780
1980 rem
1990 rem
2000 rem
2010 rem
2020 rem conversie
2030 rem
2040 print"[CLR-HOME][6xneer][10xspatie]Even Gedul
     d AUB[5xneer]"
2050 open15,8,15:open5,8,5,"#"
2060 fora=0to95:hh=hh+br(a):next
2070 poke49152,(ho-om-1):hh=int(hh/8)+1:poke49153,
     hh
2080 poke49154,0:poke49155,ho:poke49156,8:lb=0
2090 poke49157,0:poke49158,202:poke49159,0:hb=0
2100 fora=0to96:poke49160+2*a,lb:poke49161+2*a,hb
2110 lb=lb+br(a):if lb>255thenhb=hb+1:lb=lb-255
2120 nexta:d=1:as=0
2130 forc=1to ho:d=1:as=0:fora=1tohh:forb=0to7
2140 a$=mid$(a$(as,c),d,1)
2150 ifa$=""thend=1:as=as+1:goto2140
2160 if a$="*"thenby=by+2↑(7-b)
2170 d=d+1:nextb
2180 poke49353+(c-1)*hh+a,by:by=0:nexta:nextc
2190 ee=49353+ho*hh:pokeee,0:bb=49152:gosub2550
2200 fora=1toho:poke52290+2*a,0:poke52291+2*a,255:
     next
```

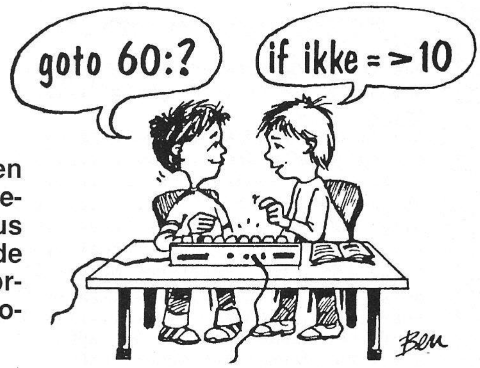
```
2210 poke52292+2*ho,tt:poke52293+2*ho,ss
2220 fora=0to125-ho:lb=52294+2*ho+2*a:pokelb,0
2230 pokelb+1,255:nexta:bb=52292:ee=52292+253
2240 gosub2550
2250 input"fontnaam";n$:open8,8,8,"0:"+n$+",p,w"
2260 fora=2to96:readb:print#8,chr$(b);:next
2270 lb=202+hh*ho:th=int(lb/256):tl=lb-th*256
2280 print#8,chr$(tl);chr$(th);
2290 fora=99to127:print#8,chr$(0);:next
2300 print#8,chr$(26);chr$(0);chr$(ho+128);chr$(5)
     ;:fora=1to30
2310 print#8,chr$(0);:nexta:close8,8,8
2320 vt=18:vs=1
2330 print#15,"b-r"5,0;vt;vs
2340 z=0:gosub2490:tv=by:gosub2490:sv=by
2350 fort=0to7:b$=""
2360 z=5+t*32:fora=1tolen(n$):gosub2490:b$=b$+chr$
     (by):nexta
2370 ifleft$(b$,len(n$))=n$then2390
2380 nextt:vt=tv:vs=sv:goto2330
2390 z=3+t*32:gosub2490:tr=by:gosub2490:sr=by
2400 print#15,"b-p:";5;(2+t*32);
2410 print#5,chr$(131);chr$(tt);chr$(ss);
2420 print#15,"b-p:";5;(21+t*32);
2430 print#5,chr$(tr);chr$(sr);
2440 fora=1to7:readb:print#5,chr$(b);:nexta
2450 print#5,chr$(te+1);:print#15,"u2:"5;0;vt;vs
2460 close5,8,5:close15,8,15:end
2470 rem
2480 rem
2490 print#15,"b-p:"5;z;:get#5,by$:by$=by$+chr$(0)
       2500 z=z+1:ifby$=chr$(0)thenby=0:goto2520
2510 by=asc(by$)
2520 return
2530 rem
2540 rem
2550 t=5:s=1:cc=0:d=int((ee-bb)/254)+1:te=te+d
2560 print#15,"b-a:"0;t;s:tt=t:ss=s
2570 input#15,a,b$,c,f:ifa=65thent=c:s=f:goto2560
2580 foraa=1tod
2590 dd=1:print#15,"u2:"5;0;t;s
2600 print#15,"b-a:"0;t;s
2610 i=i+1:ifi<>2thengoto2590
2620 i=0:input#15,a,b$,c,e:ifd=1thenc=0:e=255
2630 print#15,"b-p:"5;0;:print#5,chr$(c);chr$(e);
2640 print#5,chr$(peek(bb+cc));:cc=cc+1:dd=dd+1
2650 ifdd>254then2680
2660 if ee=bb+ccthen2690
2670 goto2640
2680 print#15,"u2:"5;0;t;s:t=c:s=e:nextaa
2690 print#15,"b-p:"5;0;:print#5,chr$(0);chr$(dd);
       2700 print#15,"u2:"5;0;t;s:return
2710 rem
2720 rem
2730 dim a$(96,ho):dim br(96)
2740 fora=1toho:a$(96,a)="........":nexta:return
2750 rem
2760 rem
2770 data3,21,191,255,255,255,128,0,1,128,0,1,128
2780 data0,1,191,128,1,152,128,9,152,0,25,152,0
2790 data25,158,0,61,152,231,153,153,182,217,153
2800 data182,217,153,182,217,188,230,205,128,0,1
2810 data128,0,1,128,0,1,128,0,1,128,0,1,128,0,1
2820 data255,255,255,131,8,1,0,0,255,255,0,0,67
2830 data111,114,121,32,32,32,32,32,32,32,32,86
2840 data49,46,49,0,0,0,0,1,8,86,4,7,12,0

*** EINDE LISTING ***
```



Na negen echte aflevering is de vaart er wat betreft het zelf maken van leerprogramma's enigszins uit. De serie zelf leerprogramma's maken wordt dan ook in dit nummer afgesloten, maar dat betekent niet dat Commodore Info nu geen aandacht meer zal besteden aan het gebruik van de Commodore modellen in het onderwijs.

Zelf Leerprogramma's maken in BASIC (Slot)



Het zelf maken van software ter ondersteuning van het leerproces is geen onmogelijke opgave. Wij hopen dat deze stelling door onze serie artikelen over dit onderwerp voldoende bewezen is. We sluiten de reeks nu af. Dus geen nieuwe programma's deze keer. Op een later tijdstip nemen we de draad wellicht weer op. We gaan dan verder met het presenteren van voorbeelden van educatieve software voor de leerkracht die door zelf te programmeren, meer uit de computer wil halen.

Vergissing

In deze laatste aflevering aandacht voor twee dingen. Ten eerste hebben wij nog iets goed te maken. In het vorige nummer van Commodore-Info zijn door een vergissing bij de lay-out de listing bij het artikel weggevallen. We zijn ze domweg vergeten in te voegen. We halen de programmalistings altijd rechtstreeks uit het geheugen van de computer. Op deze wijze weten we zeker, dat de afgedrukte listing correct is en zeker zal werken op uw computer.

Bij het opmaken van een listing als tekst, wil er nog wel eens iets misgaan. Het opmaakprogramma kent soms sommige karakters niet en zet er andere voor in de plaats. Als we werken met een rechtstreekse uitdraai van het gepresenteerde programma, kennen we die problemen niet. Echter is het dan opletten geblazen om ervoor te zorgen, dat tekst en listing samen bij de opmaakredactie aankomen. Dat ging de vorige keer even mis. Onze excuses daarvoor.

U krijgt dus alsnog de listings van de programma's van de vorige aflevering. De eerste is van de autorun utility, waarmee u een programma op diskette zelfstartend kan maken. Het tweede is van een programma, waarmee u probleemloos programma's vanaf de diskette kunt laden en automatisch laat starten.

Voor nadere details moet u de beschrijving in de vorige Commodore-Info maar eens (her)- lezen.

Alsnog veel plezier ermee.

Project

Dan even aandacht voor een tweede punt. Zoals u weet draagt Commodore-Info het onderwijs een warm hart toe. We besteden altijd ruim aandacht aan schoolse zaken als het om computers gaat.

In de komende maanden zult u ongetwijfeld meer horen over het project, dat de mensen van Sala Communications (de makers van o.a. dit blad) willen starten. Wij willen nu alvast een tipje van de sluier oplichten en u iets vertellen over dit unieke project. Doelgroep is het onderwijs in de meest brede zin des woords.

Doel is het stimuleren van computergebruik in het onderwijs. Zowel op management- als ook op klasse-nivo. Het gaat erom, dat de mensen in het onderwijsveld meer vat krijgen op het fenomeen computer. Door middel van het opzetten van informatie- en softwarebanken, het organiseren van scholing- en oriëntatiebijeenkomsten, het verschaffen van gerichte informatie, software en wellicht hardware willen we de computer de scholen binnen halen en in andere gevallen de computer een nieuw leven inblazen cq. het stof eraf blazen. Kortom, de computer moet binnen het onderwijs de plaats krijgen, die hij verdient. Om een indicatie van onze bedoelingen te geven: Als de verwachtingen aangeven, dat het nog wel vijf tot zes jaar kan duren voordat de computer een algemeen aanvaard hulpmiddel is binnen het onderwijs, dan willen wij ervoor zorgen dat dat binnen twee jaar het geval is.

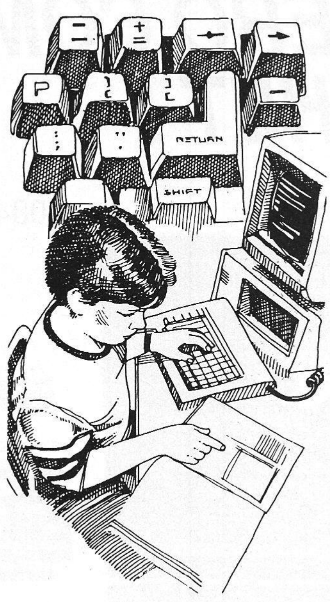
Achter de schermen wordt nu reeds hard gewerkt om alle gegevens uit te werken tot een bruikbare strategie. Een strategie die moet leiden tot een eerste aanzet van een zinvol en efficiënt gebruik van computers in het onderwijs. Binnenkort zult u hierover meer vernemen.

B. Munniksma

Listing AUTORUN

```
100 REM AUTORUN
110 :
120 FOR I = 49152 TO 49407
130 READ A:POKE I,A
140 Z=Z+A:NEXT I:PRINT Z
150 DATA 169,169,141,167,2
160 DATA 169,0,141,168,2
170 DATA 169,133,141,169,2
180 DATA 169,157,141,170,2
190 DATA 169,88,141,171,2
200 DATA 169,32,141,172,2
210 DATA 169,94,141,173,2
220 DATA 169,166,141,174,2
230 DATA 169,32,141,175,2
240 DATA 169,142,141,176,2
250 DATA 169,166,141,177,2
260 DATA 169,76,141,178,2
270 DATA 169,174,141,179,2
280 DATA 169,167,141,180,2
290 DATA 169,96,141,181,2
300 DATA 169,167,141,2,3
310 DATA 169,2,141,3,3
320 DATA 169,167,141,193,0
330 DATA 169,2,141,194,0
340 DATA 169,8,141,186,0
350 DATA 32,68,229,165,186
360 DATA 208,3,76,19,247
370 DATA 201,3,240,249,144
380 DATA 95,169,97,133,185
390 DATA 164,183,208,3,76
400 DATA 16,247,32,213,243
410 DATA 32,213,192,165,186
```

```
420 DATA 32,12,237,165,185
430 DATA 32,185,237,160,0
440 DATA 32,142,251,165,172
450 DATA 32,221,237,165,173
460 DATA 32,221,237,32,209
470 DATA 252,176,22,177,172
480 DATA 32,221,237,32,225
490 DATA 255,208,7,32,66
500 DATA 246,169,0,56,96
510 DATA 32,219,252,208,229
520 DATA 32,254,237,36,185
530 DATA 48,17,165,186,32
540 DATA 12,237,165,185,41
550 DATA 239,9,224,32,185
560 DATA 237,32,254,237,24
570 DATA 32,226,252,162,10
580 DATA 160,9,32,240,255
590 DATA 162,0,189,234,192
600 DATA 32,210,255,232,224
610 DATA 20,208,245,96,67
620 DATA 79,77,77,79,68
630 DATA 79,82,69,45,73
640 DATA 78,70,79,32,32
650 DATA 49,57,56,56,0,0
```



Listing AUTO-LADER

```
1 rem ******************************
2 rem ** leerprogramma's in basic  **
3 rem **                           **
4 rem ** auto-lader voor diskette  **
5 rem **                           **
6 rem **  (c) 1988 bob munniksma   **
7 rem **                           **
8 rem **   voor commodore-info     **
9 rem ******************************
10 :
30 rem initialiseren
35 print chr$(14)chr$(8)chr$(147);
40 poke 53281,0:poke 53280,0
45 dim p$(15):poke 646,7
50 print"Geduld.":print chr$(19);
55 k1$=chr$(176)+chr$(96)+chr$(174)
60 k2$=chr$(125)+chr$(32)+chr$(125)
65 k3$=chr$(173)+chr$(96)+chr$(189)
70 k5$=chr$(17):for i=1 to 3
75 k4$=k4$+chr$(96)
80 k5$=k5$+chr$(157):next
85 k$=k1$+k5$+k2$+k5$+k3$
90 ne$=chr$(19):for i=1 to 20:
95 ne$=ne$+chr$(17):next
100 k6$=left$(k1$,2)+k4$+k4$+right$(k1$,2)
105 k7$=left$(k3$,2)+k4$+k4$+right$(k3$,2)
110 :
115 rem directory inlezen
120 open 15,8,15,"i0":open 8,8,2,"#"
125 d=0:t=18:s=1
130 print#15,"u1";2;0;t;s
135 print#15,"b-p";2;0
140 get#8,x$:if x$="" then x$=chr$(0)
145 t=asc(x$)
150 get#8,x$:if x$="" then x$=chr$(0)
155 s=asc(x$):for x=0 to 7
160 print#15,"b-p";2;x*32+2
165 get#8,x$:if x$="" then x$=chr$(0)
170 if asc(x$)=0 then 210
175 print#15,"b-p";2;x*32+5
180 f$="":for y=0 to 15
185 get#8,x$:if x$="" then x$=chr$(0)
190 if asc(x$)=160 then 200
195 f$=f$+x$:next y
200 p$(d)=f$:f$="":d=d+1
205 if d>14 then d=14:goto 215
210 next x:if t<>0 then 130
215 close 8:close 15
220 dd=d/2:if dd<1 then 270
225 if dd<>int(dd) then dd=int(dd+1)
230 print chr$(18)"Maak een keuze:"
235 for i=1 to dd
240 print k$;chr$(145)chr$(145)tab(20)k$
245 next:for i=1 to dd
250 print left$(ne$,3*i)tab(1);
255 print chr$(i+192);tab(3)p$(i);
260 print tab(21)chr$(i+192+dd);
265 print tab(23)p$(i+dd):next:print
270 print k6$
275 print chr$(125)" RETURN "chr$(125);
280 print" andere diskette":print k7$;
285 get t$:if t$="" then 285
290 ts=asc(t$):if ts=13 then run
295 ts=ts-64:if ts<1 or ts>d then 285
300 :
305 printchr$(147)left$(ne$,13);
310 print tab(10)"Even geduld graag..."
315 print:print tab(10)chr$(18)p$(ts);
320 print chr$(146)" wordt geladen"
325 poke 646,0:print chr$(19);
330 print"load"chr$(34)p$(ts)chr$(34)",8"
335 print left$(ne$,6)"run"
340 poke 631,19:poke 632,13
345 poke 633,13:poke 198,3
350 end
```

Checksum C-64

Syntax Checksum

Het overtikken van een listing kan een heel karwei zijn en als u een beetje normaal mens bent dan maakt u daarin beslist een aantal fouten. Nu is niets moeilijker om de fouten uit je eigen werk te halen. Al geruime tijd heeft Jan Bodzinga hiervoor een zgn. Checksum-programma geschreven. Om de vele nieuwe lezers van Commodore-info te helpen volgt hieronder nog een keer een volledige uitleg over de werking van dit programma, waarmee het, hoe vreemd dat misschien ook lijkt, echt mogelijk is om met behulp van dit programma de fouten in elke door ons geplaatste listing op te sporen.
Hiervoor gaat u als volgt te werk:
1. U tikt de listing heel zorgvuldig over en SAVEt hem voordat u het programma RUNt op een diskette of cassette.
2. U tikt het RUN commando in. Mocht het programma de boodschap 'FOUT in dataregels!' geven dan heeft u een fout bij het overtikken gemaakt. Herstel dan de fout en SAVE de verbeterde versie. Mocht het programma met de boodschap 'data is weggezet checksum testen met sys...'komen dan is tot dusver alles goed. Het programma is nu in een stukje machinetaalgeheugen gezet. Als u het NEW commando geeft blijft het toch in de computer staan.
Alle door ons geplaatste programma's zijn in Basic geschreven.
Als u een programma heeft overgetikt SAVE het eerst, mocht er iets mis gaan dan hoeft u niet de gehele listing opnieuw te gaan intikken. Als u nu een programma op fouten wilt gaan controleren dan kunt u dat in het geheugen laden(wel eerst het checksum programma hebben gerund). Vervolgens typt u zonder het programma te runnen de opdracht sys 49152(c-64) of sys 1536 (c-16 en plus/4)in.
Als alles goed is gegaan loopt er nu een rij regelnummers over het scherm met getallen erachter. Dezelfde lijst staat ook achter elk door ons geplaatste programma. Wijkt nu een nummer achter een regelnummer af van het nummer dat in het blad staat dan heeft u in die regel iets anders ingetikt dan er in het blad stond. U kunt de stroom getallen d.m.v. de RUN/STOP toets pauzeren en weer vervolgen met de F1 of F7 toets. Het is uitermate belangrijk dat u goed met dit programma overweg kunt en mocht u het niet goed werkend krijgen bel dan gerust even met onze listingservice telefoonlijn. (Maandag 17.00 - 21.00 uur. Telefoonnummer 02155-25162.)

```
1      rem ******************************
       ***
  2    rem basic loader "SYNTAX.CHECKSUM"
3      rem na de commando's "run" en "new
       "
  4    rem blijft dit programma in het ge
       -
5      rem heugen. laad het te testen pro
       -
  6    rem gramma en tik daarna sys 49152
       .
  7    rem ******************************
       ***
10     i=49152  :rem beginadres
20     reada:ifa<0then40:rem data ingelez
       en
30     pokei,a:i=i+1:b=b+a:goto20
40     if b<>16844thenprint"[SHIFT-CLR]fo
       ut[SPACE]in[SPACE]dataregels!":b=0
       :end
50     poke49184,148:poke49185,192
55 i=49300
60     read a: ifa<0then80
70     pokei,a:b=a+b:i=i+1:goto60
80     if b<>20068thenprint"[SHIFT-CLR]fo
       ut[SPACE]in[SPACE]dataregels![SPAC
       E](vanaf[SPACE]regel[SPACE]240)":b
       =0:end
90     print"data[SPACE]is[SPACE]weggezet
       "
95 print"checksum[SPACE]testen[SPACE]
       met[SPACE]sys49152"
100    data 165,43,166,44,133,163,134,164
       ,169,147
110    data 32,210,255,160,0,240,3,32,73,
       192
120    data 32,73,192,208,1,96,32,225,255
       ,208
130    data 3,76,116,164,32,81,192,32,73,
       192
140    data 240,12,201,32,240,247,24,101,
       167,133
150    data 167,76,37,192,166,167,169,0,1
       32,168
160    data 32,205,189,169,13,32,210,255,
       164,168
170    data 76,17,192,200,208,2,230,164,1
       77,163
180    data 96,162,0,189,123,192,240,6,32
       ,210
190    data 255,232,208,245,32,73,192,170
       ,32,73
200    data 192,132,168,32,205,189,162,3,
       169,32
210    data 32,210,255,202,208,250,169,0,
       133,167
220    data 164,168,96,82,69,71,69,76,32,
       0
230    data  -1
240    data 165,197,201,3,240,7,201,4,240
250    data 6,76,148,192,76,34,192,169
260    data 147,32,210,255,76,161,192
270    data -1
```

** EINDE LISTING checksum 64 **

REGEL	1	249	REGEL	140	96
REGEL	2	84	REGEL	150	96
REGEL	3	105	REGEL	160	127
REGEL	4	2	REGEL	170	71
REGEL	5	246	REGEL	180	223
REGEL	6	152	REGEL	190	73
REGEL	7	249	REGEL	200	79
REGEL	10	157	REGEL	210	109
REGEL	20	64	REGEL	220	106
REGEL	30	38	REGEL	230	225
REGEL	40	57	REGEL	240	16
REGEL	50	14	REGEL	250	163
REGEL	55	251	REGEL	260	92
REGEL	60	192	REGEL	270	22
REGEL	70	42			
REGEL	80	244			
REGEL	90	245			
REGEL	95	237			
REGEL	100	183			
REGEL	110	158			
REGEL	120	232			
REGEL	130	183			

Checksum C 16

Checksum c16

```
10     rem ******************************
       ***
20     rem syntax.checksum
30     rem voor c-16 & plus/4
40     rem
50     rem syntax testen met 'sys 1536'
60     rem
70     rem v.851128.16       jan bodzinga
80     rem ******************************
       ***
90     i=1536       :rem beginadres
100    reada:ifa>=0then pokei,a:i=i+1:got
       o100
110    print"data[SPACE]is[SPACE]weggezet
       "
120    print"cheksum[SPACE]printen[SPACE]
       met[SPACE]'sys[SPACE]1536'
130    end
200    data 165, 43,166, 44,133
210    data  31,134, 32,169,147
220    data  32,210,255,160,  0
230    data 240,  3, 32, 73,  6
240    data  32, 73,  6,208,  1
250    data  96, 72,152, 32,131
260    data   6,168,104,234, 32
270    data  81,  6, 32, 73,  6
280    data 240, 12,201, 32,240
290    data 247, 24,101,252,133
300    data 252, 76, 37,  6,166
310    data 252,169,  0,132,253
320    data  32, 95,164,169, 13
330    data  32,210,255,164,253
340    data  76, 17,  6,200,208
350    data   2,230, 32,177, 31
360    data  96,162,  0,189,123
370    data   6,240,  6, 32,210
380    data 255,232,208,245, 32
390    data  73,  6,170, 32, 73
400    data   6,132,253, 32, 95
410    data 164,162,  3,169, 32
420    data  32,210,255,202,208
430    data 250,169,  0,133,252
440    data 164,253, 96, 82, 69
450    data  71, 69, 76, 32,  0
460    data   0, 72,138, 72, 32
470    data 225,255,240,251,104
480    data 170,104, 96, -1
```

** EINDE LISTING checks16

REGEL 10	249	REGEL 290	248
REGEL 20	247	REGEL 300	118
REGEL 30	121	REGEL 310	204
REGEL 40	143	REGEL 320	165
REGEL 50	75	REGEL 330	252
REGEL 60	143	REGEL 340	106
REGEL 70	8	REGEL 350	98
REGEL 80	249	REGEL 360	163
REGEL 90	103	REGEL 370	45
REGEL 100	2	REGEL 380	0
REGEL 110	245	REGEL 390	58
REGEL 120	237	REGEL 400	108
REGEL 130	128	REGEL 410	159
REGEL 200	210	REGEL 420	245
REGEL 210	208	REGEL 430	202
REGEL 220	142	REGEL 440	176
REGEL 230	1	REGEL 450	12
REGEL 240	3	REGEL 460	54
REGEL 250	157	REGEL 470	43
REGEL 260	155	REGEL 480	1
REGEL 270	215		
REGEL 280	186		



PRINT-OUT C 64 met o.a. Cset en Penalty

Cset-Designer c64

Met behulp van dit programma is het mogelijk zelf karakters te ontwerpen of de karakterset te veranderen. De desbetreffende karakterset die door de gebruiker kan worden ontworpen staat op adres $2000=#8192 Voor dit Programma treft U een zgn loader aan : 'loader cset-d' , als U dit programma runt wordt het hoofdprogramma geladen. Dit is gedaan om het starten van het programma wat makkelijker te maken, het wordt namelijk op een andere plaats in het geheugen geladen. N.L.: $4001 oftewel adres 16385.

Na het opstarten ziet U een hoofdmenu met hierin de volgende punten:

1. karakterset ontwerpen
2. karakterset wegschrijven
3. karakterset laden
4. karakterset wissen
5. Directory

1. Karakterset ontwerpen

De karakterset die op dat moment op adres 8192 staat, zal op de rechterhelft van het scherm verschijnen en het werkgebied vanzelf sprekend op de overgebleven linker helft.

U ziet dat de cursor op '@' staat. Deze cursor is door de hele tekenset te bewegen met behulp van de cursortoetsen. Als de cursor op het teken staat dat U wilt gaan veranderen, drukt U op return.

Tijdens het bewerken van een teken is de tekenset op de rechterhelft veranderd in de normale karakterset van de commodore 64. Sommige karaktersets hebben wel eens een ander teken staan op de plaats van de tekens waar de cursor uit het werkgebied uit bestaat. N.l. de schermcodes: 76, 79, 80, 122 en nog een aantal tekens die de lijnen om het werkgebied vormen.

Het teken van uw keuze zal vergroot in het omlijnde werkgebied verschijnen. Hierna ziet U in de linkerbovenhoek de cursor.

De toetsen die tijdens het bewerken zijn te gebruiken zijn:

s = teken onderste boven
r = tekens kopieren naar andere schermccode's en deze worden tegelijk de inverse karakters van deze te kopieren tekens.
q = quit (de tekst wordt gewist en er wordt teruggekeerd naar het hoofdmenu.
 = teken in de tekenset plaatsen
- = teken negatief
spatie = verwijder bit
home = cursor gaat naar de uitgangspositie
/home = het teken wordt gewist
= een bit wordt geplaatst
naar links = hoofdmenu

En bij deze toetsen worden natuurlijk ook de cursor toetsen normaal gebruikt.

2. Karakterset saven.

Na het invoeren van de Filenaam zal de karakterset geheel naar disk worden gesaved. Deze set kan weer worden geladen met: load "naam",8,1 U kunt hem dan weer te zien krijgen door het intypen van: poke 53272,24

3. Karakterset laden.

Als U de juiste filenaam invoert, zal de karakterset die U bijvoorbeeld nog niet af heeft, worden ingeladen. U kunt nu weer doorgaan door in het hoofdmenu een "1" in te toetsen. Er kunnen ook karaktersets worden ingeladen die niet met dit programma zijn gemaakt als ze maar op het adres $2000 zijn weggeschreven.

4. Karakterset wissen.

Het wissen van een karakterset is erg eenvoudig, gewoon op q drukken en alles is verdwenen.

5. Directory.

Het is een directory waar alleen de namen op het scherm worden weergegeven. Maar U kunt wel een begin en een eindadres van een file te weten komen. U handeld dan als volgt: Als we een directory, of een gedeelte daarvan op het scherm krijgen dan drukt U op return. Er zal dan een pijl verschijnen achter de eerste filenaam. Met de cursortoetsen kunt U deze pijl bewegen tot hij achter de juiste file staat. U drukt dan nogmaals op de returntoets en het startadres, zowel hexadecimaal als decimaal verschijnt achter de pijl. Indien U nu terug wilt naar het hoofdmenu drukt U op de spatiebalk.

Door al deze informatie zouden we haast de naam van al dit schoons vergeten te vermelden. Het is de Heer P.J. van der Mespel uit Den Haag.

```
10    dimz(64),n$(100)
20    r=49152:al=r+28:as=r+68
30    r$=chr$(13):o$=chr$(0):b=53280:sc=53272:hx$="0123456789abcdef"
40    s$="[40xSPACE]"
50    t(1)=0:t(2)=1:t(3)=40:t(4)=41
60    s(1)=79:s(2)=80:s(3)=76:s(4)=122
70    fori=0to22:readq:poke52992+i,q:next
80    fori=1to8:readvb(i):next
90    fori=rto49263:readj:pokei,j:next
100   rem kopieer cset naar $2000
110   poke56334,peek(56334)and254
120   poke1,peek(1)and251:sys52992
130   poke1,peek(1)or4
140   poke56334,peek(56334)or1
150   rem hoofdmenu
160   printchr$(142)chr$(8)chr$(147);
170   pokeb,11:pokeb+1,11:pokesc,21
180   ford=1to24
190   print"[CTRL-9][CTRL-2][SPACE][COM-8][SPACE][COM-5][SPACE][COM-4][34xSPACE][COM-5][SPACE][COM-8][SPACE][CTRL-2][SPACE]";
200   nextd
210   print"[HOME][CTRL-2][CTRL-9]"s$"[SPACE][COM-8]"left$(s$,38)"[CTRL-2][2xSPACE][COM-8][SPACE][COM-5]"left$(s$,36)
220   print"[HOME]"tab(10)"[4xCRSR-DOWN][COM-8]h[SPACE]o[SPACE]o[SPACE]f[SPACE]d[SPACE]m[SPACE]e[SPACE]n[SPACE]u[CRSR-DOWN]"
```

```
230  printtab(8)"[CRSR-DOWN]1.[SPACE]karakterset[SPACE]ontwerpen"
240  printtab(8)"[CRSR-DOWN]2.[SPACE]karakterset[SPACE]wegschrijven"
250  printtab(8)"[CRSR-DOWN]3.[SPACE]karakterset[SPACE]laden"
260  printtab(8)"[CRSR-DOWN]4.[SPACE]karakterset[SPACE]wissen"
270  printtab(8)"[CRSR-DOWN]5.[SPACE]directory"
280  print"[5xCRSR-DOWN][CTRL-9][CTRL-2][SPACE][COM-8][SPACE][COM-5]"left$(s$,36)"[COM-8][SPACE][CTRL-2][2xSPACE][COM-8]"left$(s$,38)"[CTRL-2]"s$
290  print"[6xCRSR-UP]"tab(8)"[COM-8]toets[SPACE]uw[SPACE]keuze[SPACE]>[SPACE][CRSR-LEFT]";
300  poke204,0:getk$:ifk$<"1"ork$>"5"then300
310  ke=val(k$):poke204,1:onkegoto320,1390,1390,1510,1660
320  rem karakterset ontwerpen
330  pokesc,24
340  pokeb,0:pokeb+1,0:print"[COM-6][SHIFT CLR]";
350  print"[4xCRSR-DOWN][CRSR-RIGHT][16xCOM-I]"
360  ford=1to16:print"[COM-L][16xSPACE][COM-K]":nextd
370  print"[CRSR-RIGHT][16xCOM-U][HOME]"
380  p=1090:c=-1:q=p
390  forn=qtoq+10:ifn>2012then410
400  c=c+1:poken,c:nextn:q=q+40:goto390
410  x=p:pokex,128
420  geta$:ifa$=""then420
430  ifa$=r$thenpokesc,21:goto560
440  ifa$="[CRSR-UP]"theny=-40:goto510
450  ifa$="[CRSR-DOWN]"theny=40:goto510
460  ifa$="[CRSR-RIGHT]"theny=1:goto510
470  ifa$="[CRSR-LEFT]"theny=-1:goto510
480  ifa$="r"thenpokesc,21:gosub1280
490  ifa$="_"then160
500  goto420
510  x=x+y:ifx<porx>2012thenx=x-y:goto420
520  c%=x/40-27:ifx<(p+c%*40)thenx=p+(c%-1)*40+10:y=-30
530  ifx>(p+c%*40+10)thenx=p+(c%+1)*40:y=30
540  pokex,(peek(x)+128)and255:
           pokex-y,(peek(x-y)+128)and255
550  goto420
560  pokex,(peek(x)+128)and255:gosub950
570  c%=x/40-27:cn=x-(p+c%*40)+c%*11
580  ch=1225:fora=0to7:forf=7to0step-1
590  ifpeek(cn*8+8192+a)and2^fthengosub610
600  ch=ch+2:nextf:ch=ch+64:nexta:p=1225:x=p:goto620
610  fore=1to4:pokech+t(e),160:nexte:return
620  fore=1to4:g=0:ifpeek(p+t(e))and128theng=128
630  pokep+t(e),s(e)+g:nexte
640  geta$:ifa$=""then640
650  ifa$="_"then160
660  ifa$="r"thengosub1280:pokesc,21
670  ifa$="s"thengosub1220
680  ifa$="q"then1170
690  ifa$=r$then1020
700  ifa$="-"thengosub1130
710  ifa$="[SPACE]"then930
720  ifa$="[HOME]"then990
730  ifa$="[SHIFT-CLR]"thengosub950:goto990
740  ifa$=chr$(133)then970
750  ifa$="[CRSR-UP]"theny=-80:goto800
760  ifa$="[CRSR-DOWN]"theny=80:goto800
770  ifa$="[CRSR-RIGHT]"theny=2:goto800
780  ifa$="[CRSR-LEFT]"theny=-2:goto800
790  goto640
800  x=x+y:ifx<porx>1800thenx=x-y:goto640
810  d%=(x-p)/80
820  ifp+d%*80+14<xthenx=x-y
830  ifx<p+d%*80thenx=x-y
840  fore=1to4:g=0
850  ifpeek(x+t(e))and128theng=128
860  pokex+t(e),s(e)+g
870  nexte
880  fore=1to4:g=32
890  ifpeek(x-y+t(e))and128theng=160
900  pokex-y+t(e),g
910  nexte
920  goto640
930  rem spatie
940  fore=1to4:pokex+t(e),32:nexte:y=2:goto800
950  rem clr-char
960  print"[HOME][5xCRSR-DOWN]";:forcl=1to16:print"[CRSR-RIGHT][16xSPACE]":nextcl:return
970  rem plaats bit
980  fore=1to4:pokex+t(e),160:nexte:y=2:goto800
990  rem cursor > home
1000 fore=1to4:g=32:ifpeek(x+t(e))and128theng=160
1010 pokex+t(e),g:nexte:x=p:goto620
1020 print"[HOME]plaats[SPACE]karakter[SPACE]op[SPACE]nummer"cn"?"
1030 getj$:ifj$<>"j"andj$<>"n"then1030
1040 ifj$="j"then1080
1050 print"[HOME]"s$"[HOME]voer[SPACE]scherm-code[SPACE]in[SPACE]>";:l=3:gosub1530
1060 cn=val(f$):ifcn=>0andcn<256then1080
1070 print"[HOME]"s$:goto1050
1080 pokesc,24
1090 foral=0to7:v=0:forw=7to0step-1
1100 ifpeek(1225+al*80+(7-w)*2)and128thenv=v+2^w
1110 nextw:pokecn*8+8192+al,v:nextal
1120 print"[HOME]"s$:p=1090:gosub950:goto410
1130 rem negatief
1140 fore=0to7:forf=0to7:fort=1to4
1150 poke1225+e*80+f*2+t(t),(peek(1225+e*80+f*2+t(t))+128)and255
1160 nextt,f,e:return
1170 rem quit
1180 print"[HOME]weet[SPACE]u[SPACE]het[SPACE]zeker[SPACE]?"
1190 getj$:ifj$<>"j"andj$<>"n"then1190
1200 ifj$="j"thenrun
1210 print"[HOME]"s$:pokesc,24:goto640
1220 rem ondersteboven
1230 fore=1to4:g=32:ifpeek(x+t(e))and12
```

```
8theng=160
1240 pokex+t(e),g:nexte
1250 t=0:fore=0to7:forf=0to7:z(t)=peek(
     1225+e*80+2*f):t=t+1:nextf,e:gosub
     950
1260 t=0:fore=0to7:forf=0to7:forh=1to4
1270 poke1799-e*80-2*f+t(h),z(t):nexth:
     t=t+1:nextf,e:return
1280 rem inverse copieeren
1290 print"[HOME]"s$"[HOME]vanaf[SPACE]
     welk[SPACE]karakter[SPACE](schermc
     ode)?";:l=3:gosub1530:c1=val(f$)
1300 ifc1<0orc1>255then1290
1310 print"[HOME]"s$"[HOME]t/m[SPACE]we
     lke[SPACE]schermcode[SPACE]?";:gos
     ub1530:c2=val(f$)
1320 ifc2<0orc2>255orc2<c1then1310
1330 print"[HOME]"s$"[HOME]naar[SPACE]w
     elke[SPACE]schermcode[SPACE]?";:go
     sub1530:c3=val(f$):print"[HOME]"s$
1340 ifc3<0orc3>255then1330
1350 pokesc,24:forcp=c1toc2
1360 forkc=0to7
1370 poke8192+8*c3+kc,255-peek(8192+8*c
     p+kc)
1380 nextkc:c3=c3+1:nextcp:return
1390 rem cset laden/saven
1400 print"[SHIFT-CLR][3xCRSR-DOWN]file
     naam:";:l=16:gosub1530:ll=len(f$):
     print
1410 open15,8,15:close15:ifst=-128then1
     60
1420 open15,8,15,"i0":input#15,a,b$:clo
     se15
1430 ifathenprint"[CRSR-DOWN]"b$:print"
     [CRSR-DOWN]druk[SPACE]op[SPACE]een
     [SPACE]toets...":poke198,0:wait198
     ,1:goto160
1440 poker+18,ll
1450 fort=1toll:poke(r+t+1),asc(mid$(f$
     ,t,1)):nextt
1460 poker+19,0:poker+20,32
1470 on(ke-1)goto1490,1480
1480 sys al:goto160
1490 ll=ll+2:poker+18,ll
1500 poker+21,255:poker+22,39:sys as:go
     to160
1510 run:rem karakterset gewist
1520 rem get-subroutine
1530 t=0:c$="":f$=""
1540 getc$:ifc$=""then1540
1550 fors=1to8
1560 ifasc(c$)=vb(s)then1540
1570 nexts
1580 ifc$=chr$(20)then1620
1590 ifc$=r$thenreturn
1600 t=t+1:ift>lthent=l:goto1540
1610 printc$;:f$=f$+c$:goto1540
1620 t=t-1:ift<0thent=0:goto1540
1630 f$=left$(f$,t)
1640 print"[CRSR-LEFT][SPACE][CRSR-LEFT
     ]";:goto1540
1650 rem directory
1660 open15,8,15,"i0":open2,8,2,"#"
1670 t=18:s=1:tl=1:m$="":d0=-1
1680 print#15,"b-r";2;0;t;s:print#15,"b
     -p";2;0:get#2,x$:ifx$=""thenx$=o$
1690 t=asc(x$):get#2,x$:ifx$=""thenx$=o
     $
1700 s=asc(x$):forx=0to7
1710 print#15,"b-p";2;x*32+5:m$=""
1720 fory=0to15:get#2,x$
1730 ifx$=""thenx$=o$
1740 ifasc(x$)=160then1770
1750 m$=m$+x$
1760 nexty
1770 ifm$="..."then1810
1780 foru=1tolen(m$)
1790 ifasc(mid$(m$,u,1))thenn$(tl)=m$:t
     l=tl+1:m$="":goto1810
1800 nextu
1810 iftl>100then1830
1820 nextx:ift<>0then1680
1830 close2:pokesc,23
1840 print"[SHIFT-CLR]Inhoud[SPACE]van[
     SPACE]deze[SPACE]diskette:[HOME][C
     RSR-DOWN]"
1850 forh=1totl
1860 printn$(h):ifh/21=int(h/21)thend0=
     d0+1:gosub1880
1870 nexth:d0=d0+1
1880 geta$:ifa$=""then1880
1890 ifa$=r$thenp=1121:co=1+d0*21:goto1
     930
1900 ifa$="[SPACE]"andh<tlthenprint"[SH
     IFT CLR][CRSR-DOWN]":return
1910 ifa$="[SPACE]"thenforcl=1totl:n$(c
     l)="":nextcl:close15:goto160
1920 goto1880
1930 pokep,31
1940 getb$:ifb$=""then1940
1950 ifb$="[CRSR-DOWN]"theny=40:goto200
     0
1960 ifb$="[CRSR-UP]"theny=-40:goto2000
1970 ifb$=r$then2040
1980 ifb$="[SPACE]"thenpokep,32:a$="[SP
     ACE]":goto1900
1990 goto1940
2000 p=p+y:ifp>1921orp<1121thenp=p-y:go
     to1930
2010 co=((p-1)/40)-27+21*d0
2020 ifn$(co)=""thenco=co-1:p=p-y:goto1
     930
2030 pokep-y,32:goto1930
2040 open2,8,2,n$(co)
2050 get#2,l$,h$:close2
2060 sy=asc(h$+o$)*256+asc(l$+o$):he$="
     ":s1=sy
2070 fordt=3to0step-1
2080 d%=sy/16^dt:sy=sy-d%*16^dt
2090 he$=he$+mid$(hx$,d%+1,1)
2100 nextdt
2110 input#15,a,b$:ifathenpoke214,co-21
     *d0:print:printtab(20);b$:goto1930
2120 poke214,co-21*d0:print:printtab(20
     )"#"s1"=[SPACE]$"he$
2130 goto1930
2140 rem mt-kopieerroutine
2150 data160,8,162,0,189,0,208,157,0,32
     ,232,208,247,238,6,207,238,9,207,1
     36
2160 data208,238,96
2170 rem waarden bij get-subroutine
2180 data17,29,19,145,157,147,141,148
2190 rem mt-laden & schrijven
2200 data64,58,32,32,32,32,32,32,32,32,
     32,32,32,32,32,32
2210 data32,32,16,1,8,1,8,0,66,38,67,32
     ,162,8,169,8
2220 data160,0,32,186,255,173,18,192,16
     2,2,160,192,32,189,255,169
2230 data0,174,19,192,172,20,192,32,213
```

```
     ,255,142,21,192,140,22,192
2240 data76,108,192,0,174,19,192,134,25
     3,172,20,192,132,254,162,8
2250 data169,8,160,1,32,186,255,173,18,
     192,162,0,160,192,32,189
2260 data255,169,253,174,21,192,172,22,
     192,32,216,255,32,231,255,96
```

** EINDE LISTING cset-designer **

```
regel 10    27
regel 20    210
regel 30    150
regel 40    109
regel 50    254
regel 60    170
regel 70    206
regel 80    56
regel 90    209
regel 100   213
regel 110   42
regel 120   98
regel 130   28
regel 140   193
regel 150   52
regel 160   135
regel 170   217
regel 180   178
regel 190   49
regel 200   198
regel 210   37
regel 220   246
regel 230   84
regel 240   34
regel 250   248
regel 260   110
regel 270   10
regel 280   7
regel 290   251
regel 300   178
regel 310   226
regel 320   146
regel 330   191
regel 340   253
regel 350   94
regel 360   33
regel 370   141
regel 380   100
regel 390   151
regel 400   166
regel 410   74
regel 420   99
regel 430   217
regel 440   145
regel 450   102
regel 460   63
regel 470   106
regel 480   45
regel 490   131
regel 500   31
regel 510   105
regel 520   216
regel 530   89
regel 540   16
regel 550   31
regel 560   204
regel 570   250
regel 580   3
regel 590   28
regel 600   174
regel 610   253
regel 620   37
regel 630   130
regel 640   103
regel 650   131
regel 660   45
regel 670   50
regel 680   167
regel 690   130
regel 700   12
regel 710   41
regel 720   66
regel 730   176
regel 740   152
regel 750   151
regel 760   108
regel 770   66
regel 780   109
regel 790   35
regel 800   113
regel 810   212
regel 820   41
regel 830   26
regel 840   228
regel 850   15
regel 860   137
regel 870   199
regel 880   25
regel 890   15
regel 900   250
regel 910   199
regel 920   35
regel 930   85
regel 940   162
regel 950   187
regel 960   255
regel 970   51
regel 980   212
regel 990   212
regel 1000   94
regel 1010   230
regel 1020   145
regel 1030   178
regel 1040   169
regel 1050   15
regel 1060   172
regel 1070   240
regel 1080   191
regel 1090   118
regel 1100   218
regel 1110   105
regel 1120   42
regel 1130   210
regel 1140   11
regel 1150   51
regel 1160   129
regel 1170   210
regel 1180   27
regel 1190   185
regel 1200   106
regel 1210   189
regel 1220   109
regel 1230   94
regel 1240   247
regel 1250   145
regel 1260   111
regel 1270   180
regel 1280   69
regel 1290   53
regel 1300   198
regel 1310   206
regel 1320   13
regel 1330   194
regel 1340   197
regel 1350   76
regel 1360   204
regel 1370   90
regel 1380   218
regel 1390   206
regel 1400   89
regel 1410   221
regel 1420   71
regel 1430   169
regel 1440   192
regel 1450   246
regel 1460   25
regel 1470   158
regel 1480   133
regel 1490   184
regel 1500   77
regel 1510   103
regel 1520   172
regel 1530   103
regel 1540   155
regel 1550   147
regel 1560   104
regel 1570   213
regel 1580   142
regel 1590   79
regel 1600   209
regel 1610   153
regel 1620   156
regel 1630   31
regel 1640   223
regel 1650   68
regel 1660   68
regel 1670   121
regel 1680   186
regel 1690   240
regel 1700   104
regel 1710   190
regel 1720   158
regel 1730   69
regel 1740   221
regel 1750   186
regel 1760   219
regel 1770   237
regel 1780   226
regel 1790   110
regel 1800   215
regel 1810   224
regel 1820   253
regel 1830   202
regel 1840   127
regel 1850   240
regel 1860   133
regel 1870   121
regel 1880   158
regel 1890   242
regel 1900   32
regel 1910   43
regel 1920   90
regel 1930   119
regel 1940   157
regel 1950   147
regel 1960   190
regel 1970   134
regel 1980   40
regel 1990   87
regel 2000   237
regel 2010   100
regel 2020   79
regel 2030   12
regel 2040   20
regel 2050   54
regel 2060   146
regel 2070   87
regel 2080   210
regel 2090   106
regel 2100   26
regel 2110   181
regel 2120   47
regel 2130   86
regel 2140   146
regel 2150   64
regel 2160   129
regel 2170   131
regel 2180   250
regel 2190   147
regel 2200   116
regel 2210   156
regel 2220   95
regel 2230   67
regel 2240   87
regel 2250   250
regel 2260   194
```

Programma Disk redder c64

Austin Putman, één van onze jongere lezers, heeft een programmaatje geschreven dat een per ongeluk van de schijf gewist programma weer terug kan halen. Het gaat echter alleen als dit gebeurd is door het commando: SCRATCH.

```
1     rem disk redder 64
2     rem door austin putman
3     rem
4     rem
10    poke53272,23
20    gosub560:re%=0
30    print"[SHIFT-CLR][CTRL-2]Ik[SPACE]
      bekijk[SPACE]diskette[SPACE]nu."di
      %",1[CRSR-DOWN]"
40    sp%=1:tr%=di%:open15,8,15,"i0
50    open8,8,2,"#2":fi%=0
60    print#15,"u1:"2;0;tr%;sp%:gosub500
70    fori=0to1:get#8,a$:gosub470:p%(i)=
      asc(a$):next:a=2:gosub490
80    get#8,a$:gosub470:a=asc(a$)and7
90    h$="":fori=0to1:get#8,a$:gosub470:
      l%(i)=asc(a$):next
100   fori=3to18:get#8,a$:h$=h$+a$:next:
110   ifh$=""then130
120   print ki$(a);"[SPACE]";co$;h$;tab(
      11)co$;;l%(0);l%(1):ifa=0thengoto3
      00
130   fi%=fi%+1:iffi%<8thena=fi%*32+2:go
      sub490:goto80
140   ifp%(0)=0then160
150   print"[CRSR-DOWN]Spoorverandering"
      ;:tr%=p%(0):sp%=p%(1):fi%=0:printt
      r%;sp%"[CRSR-DOWN]":goto60
160   close8:print"[CTRL-9][CRSR-DOWN]Ca
      talog[SPACE]Einde[SPACE][CTRL-0]",
      :ifre%thenprint"[3xSPACE]Haal[SPAC
      E]Drive[SPACE]1":print#15,"v0
170   gosub500:close15:end
180   rem** retrieve a file **
190   print#15,"u1:"2;0;l%(0);l%(1):retu
      rn
200   d%=2:print"[CRSR-DOWN]Spoor,Sector
      ";
210   gosub190:fori=0to1:get#8,a$:gosub4
      70:z%(i)=asc(a$)
220   printz%(i)"[CRSR-LEFT][SPACE]";:ne
      xt:print:print"[CRSR-UP]Spoor,Sect
      or";
230   ifz%(0)=0thenprint:return
240   print#15,"b-f"0;z%(0);z%(1):l%(0)=
      z%(0):l%(1)=z%(1):z%(d%)=z%(0)
250   z%(d%+1)=z%(1):d%=d%+2
260   ifds=0then210
270   print:print"[CTRL-9]Het[SPACE]prog
      ramma[SPACE]is[SPACE]teruggeschrev
      en[5xSPACE][CTRL-0]!":x%=not(x%):a
      =fi%*32+2
280   fori=2tod%-4step2:print#15,"b-f:"0
      ;z%(i);z%(i+1):next
290   l%(0)=tr%:l%(1)=sp%:gosub190:goto4
      90
300   print"Terughalen?[SPACE](J/N)[4xSH
      IFT SPACE]
310   get a$:if a$="" then 310
320   if a$<>"j"then130
330   x%=0:print"[CRSR-DOWN][3xSPACE]Blo
      cks[SPACE]terughalen...":z%(0)=l%(
      0):gosub200:ifx%then130
340   print"[CRSR-DOWN]Is[SPACE]dit[SPAC
      E]een[SPACE][CTRL-9]P[CTRL-0]rogra
      mma,[CTRL-9]S[CTRL-0]equentieel,[S
      PACE][CTRL-9]U[CTRL-0]ser,"
350   print"of[SPACE][CTRL-9]R[CTRL-0]el
      atieve[SPACE]file.[2xSPACE]";
360   get ja$:ifja$<>"p"andja$<>"s"andja
      $<>"u"andja$<>"r"then360
370   print:print
380   print:ifja$="p"thenprint"[CTRL-9]P
      rogramma[CTRL-0]-":ki%=130:goto420
390   ifja$="u"thenprint"[CTRL-9]User[CT
      RL 0]-":ki%=131:goto420
400   ifja$="r"thenprint"[CTRL-9]Relatie
      f[SPACE][CTRL-0]-":ki%=132:goto420
410   print"[CTRL-9]Sequentieel.[SPACE][
      CTRL-0]-":ki%=129
420   print"File[SPACE]terughalen[SPACE]
      ..."
430   l%(0)=tr%:l%(1)=sp%:gosub190
440   a=fi%*32+2:gosub490:print#8,chr$(k
      i%);
450   bl%=((d%-2)/2)+1:print"[CRSR-DOWN]
      "bl%"Blocks[2xSPACE]";
460   print#15,"u2:"2;0;tr%;sp%:print"Da
      ar[SPACE]is[SPACE]ie[SPACE]weer[3x
      SPACE]![CRSR-DOWN]":re%=1:goto130
470   ifa$=""thena$=chr$(0)
480   return
490   print#15,"b-p:"2;a
500   input#15,en%,em$,et%,es%:ifen%=0th
      enreturn
510   print"[2xCRSR-DOWN][CTRL-9]disk[SP
      ACE]fout[CTRL-0]
520   print"[CRSR-DOWN]Fout[SPACE]meldin
      g[2xSPACE][CTRL-0]:[SPACE]"en%"[SP
      ACE]"em$","et%","es%
530   print"A.U.B[SPACE]'CONT'[SPACE]typ
      en.
540   end
550   return
560   dimp%(1),l%(1),z%(300),ki$(4):co$=
      chr$(34)
570   fori=0to4:readki$(i):next
580   data"[CTRL-9]DEL[CTRL-0]","SEQ","P
      RG","USR","REL"
590   pg$="[CTRL-2][SHIFT-CLR][CTRL-9][6
      xSPACE]Disk[SPACE]file[SPACE]redde
      r[SPACE]VC1540/1541[6xSPACE][CTRL-
      0]"
600   printpg$"[CRSR-DOWN]":print"Dit[SP
      ACE]programma[SPACE]maakt[SPACE]he
      t[SPACE]mogellijk[SPACE]om"
610   print"verloren[SPACE]programma's[S
      PACE]terug[SPACE]te[SPACE]halen.
620   print"U[SPACE]drukt[SPACE]op[SPACE
      ]een[SPACE]knop[SPACE]en[SPACE]de[
      SPACE]rest[SPACE]gaat[SPACE]van-";
630   print"zelf.[SPACE]Bijna[SPACE]alle
      [SPACE]programma's
640   print"[CRSR-DOWN]zijn[SPACE]100%[S
      PACE]terug[SPACE]te[SPACE]halen[SP
      ACE]als[SPACE]ze[SPACE]door[2xSPAC
      E]"
650   print"'SCRATCH'[SPACE]waren[SPACE]
      weggehaald.
660   print"[CRSR-DOWN]Bij[SPACE]het[SPA
      CE]zoeken[SPACE]verliest[SPACE]u[S
      PACE]geen[SPACE]data"
670   print"[3xCRSR-DOWN][2xSPACE]DRUK[S
```

```
        HIFT SPACE]OP[SHIFT-SPACE]EEN[SHIF
        T SPACE]KNOP[SPACE]!"
680     get td$:if td$="" then 680
690     di%=18:return
700     print"[4xCRSR-UP]":goto670
```

** EINDE LISTING programma redder **

regel 1	218	regel 340	220
regel 2	108	regel 350	192
regel 3	126	regel 360	8
regel 4	143	regel 370	108
regel 10	43	regel 380	96
regel 20	0	regel 390	44
regel 30	203	regel 400	55
regel 40	16	regel 410	225
regel 50	46	regel 420	246
regel 60	48	regel 430	119
regel 70	160	regel 440	58
regel 80	126	regel 450	159
regel 90	117	regel 460	67
regel 100	76	regel 470	236
regel 110	40	regel 480	142
regel 120	90	regel 490	21
regel 130	35	regel 500	23
regel 140	161	regel 510	234
regel 150	184	regel 520	133
regel 160	2	regel 530	175
regel 170	28	regel 540	128
regel 180	254	regel 550	142
regel 190	238	regel 560	181
regel 200	63	regel 570	83
regel 210	140	regel 580	47
regel 220	207	regel 590	203
regel 230	117	regel 600	80
regel 240	173	regel 610	254
regel 250	97	regel 620	175
regel 260	62	regel 630	60
regel 270	54	regel 640	26
regel 280	123	regel 650	5
regel 290	215	regel 660	236
regel 300	1	regel 670	118
regel 310	97	regel 680	25
regel 320	29	regel 690	149
regel 330	242	regel 700	12;



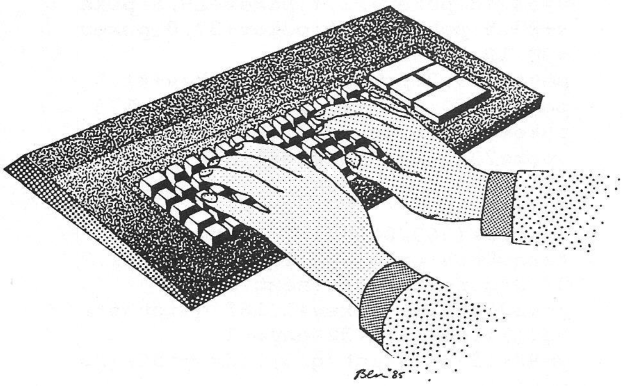
Penalty c64

Penalty is een reaktiespel voor één persoon. Met de joystick in poort twee bestuurt U de keeper met de bedoeling om zo veel mogelijk ballen tegen te houden. De keeper duikt in de richting waar de joystick heen gestuurd wordt, dat zijn er 6. Bij het programma wordt gebruik gemaakt van bijna 50 sprites. Het programma is voor de onlangs gehouden listing-wedstrijd ingestuurd en is gemaakt door Wilco Vertegaal uit Wouwse Plantage.

```
1       print"[SHIFT-CLR][8xCRSR-DOWN][CRS
        R-RIGHT]het[SPACE]inlezen[SPACE]va
        n[SPACE]de[SPACE]data[SPACE]duurt[
        SPACE]ca.[SPACE]3.20[SPACE]minute
        n.[SPACE]u[SPACE]kunt[SPACE]nu[SPA
        CE]";
2       print"nog[SPACE]even[SPACE]ontspan
        nen.":forq=192to211:gosub50:next
3       forq=0to18:fort=0to60step3:forr=0t
        o2:w=0:p=peek((193+q)*64+t+r)
4       if(pand1)=1thenw=w+64
5       if(pand2)=2thenw=w+128
6       if(pand4)=4thenw=w+16
7       if(pand8)=8thenw=w+32
8       if(pand16)=16thenw=w+4
9       if(pand32)=32thenw=w+8
10      if(pand64)=64thenw=w+1
11      if(pand128)=128thenw=w+2
12      poke((212+q)*64+t-r+2),w:nextr,t,q
        :forq=231to239:gosub50:next
13      print"[SHIFT-CLR]";:poke53280,0:po
        ke53281,13
14      a$="[COM-3]QQQQQQQQQQQQQQQQQQQQQ
        QQQQQQQQQQQQQQQQQQQ":fort=0to4:print
        a$;:next
15      print"QQQQQQQQQQQ[CTRL-9][CTRL-3][
        SPACE][CTRL-2][SPACE][CTRL-3][SPAC
        E][CTRL-2][SPACE][CTRL-3][SPACE][C
        TRL 2][SPACE][CTRL-3][SPACE][CTRL-
        2][SPACE][CTRL-3][SPACE][CTRL-2][S
        PACE][CTRL-3][SPACE][CTRL-2][SPACE
        ][CTRL-3][SPACE][CTRL-2][SPACE][CT
        RL 3][SPACE][CTRL-2][SPACE][CTRL-3
        ][SPACE][CTRL-2][SPACE][COM-3][CTR
        L 0]QQQQQQQQQQQ";
16      print"QQQQQQQQQQQ[CTRL-9][CTRL-2][
        SPACE][CTRL-0]MVVVVVVVVVVVVVVN[CTR
        L 9][CTRL-3][SPACE][CTRL-0][COM-3]
        QQQQQQQQQQQ";
17      print"QQQQQQQQQQQ[CTRL-9][CTRL-3][
        SPACE][CTRL-2][CTRL-0]V[SHIFT--]VV
        VVVVVVVVVV[SHIFT--]V[CTRL-9][SPACE
        ][CTRL-0][COM-3]QQQQQQQQQQQ";
18      print"QQQQQQQQQQQ[CTRL-9][CTRL-2][
        SPACE][CTRL-0]V[SHIFT--]VVVVVVVVVV
        VV[SHIFT--]V[CTRL-9][CTRL-3][SPACE
        ][CTRL-0][COM-3]QQQQQQQQQQQ";
19      print"[CTRL-1][CTRL-9][SPACE]commo
        dore[SPACE][CTRL-3][SPACE][CTRL-2]
        [CTRL-0]V[SHIFT--]VVVVVVVVVVVV[SHI
        FT -]V[CTRL-9][SPACE][CTRL-1][SPAC
        E]commodore[SPACE]";
20      print"[3xSPACE]info[4xSPACE][CTRL-
        2][SPACE][CTRL-0]N[14xCOM-T]M[CTRL
         9][CTRL-3][SPACE][CTRL-1][4xSPACE
        ]info[3xSPACE]";
21      printspc(12)"[CTRL-2][CTRL-0][16xC
        OM T]":print"[9xCRSR-DOWN][19xCRSR
```

```
     -RIGHT][2xSHIFT-*][CTRL-6]"
22   gs=0:forw=0to4:ks=0:kd=0
23   v=53248:pokev+21,7:pokev+29,5:poke
     v+23,5:pokev+28,5:pokev+37,0:pokev
     +38,10
24   pokev+39,7:pokev+40,15:pokev+41,5:
     pokev,160:pokev+1,206:pokev+2,175
25   pokev+3,206:pokev+4,160:pokev+5,97
     :poke2040,231:poke2041,239:poke204
     2,192
26   z=w:gosub55
27   if(peek(56320)and16)<>0then27
28   forq=0to5:poke2040,232+q:pokev+1,2
     03-3*q:fort=0to60:next:next
29   poke2040,231:pokev+1,185:q=int(6*r
     nd(1)):p=1:ifq<3thenp=-1
30   y=98+12*(q-3*int(q/3)):dx=p*55/(20
     6-y)
31   pokev+3,peek(v+3)-5:pokev+2,175+dx
     *(206-peek(v+3)):ifpeek(v+3)<=ythe
     n42
32   ifks>0then37
33   p=peek(56320):if(pand4)=4and(pand8
     )=8then41
34   rd=1:if(pand4)=0thenrd=0
35   dr=1:if(pand1)=0thendr=2
36   if(pand2)=0thendr=0
37   ks=ks+1:ifks>4then39
38   poke2042,192+ks+19*rd:pokev+4,peek
     (v+4)+8*(rd-.5):goto41
39   kd=kd+1:ifkd>5then41
40   poke2042,196+kd+19*rd+dr*5:pokev+4
     ,peek(v+4)+8*(rd-.5)
41   goto31
42   if3*rd+2-dr=qandkd>3thengs=gs+1:a=
     peek(v+2):b=peek(v+3):goto44
43   poke2040,238:fort=0to15:pokev+1,18
     5-t:forr=0to20:next:next:goto45
44   fort=0to30:pokev+2,a+t*dx:pokev+3,
     b+t:next:goto46
45   fort=0to15:pokev+1,170+t:forr=0to2
     0:next:next
46   z=w+1:gosub55
47   foret=0to1000:next:next
48   if(peek(56320)and16)<>0then48
49   goto13
50   reada$,b$:fort=0to33
51   pokeq*64+t,(asc(mid$(a$,t*2+1,1))-
     65)*16+asc(mid$(a$,t*2+2,1))-65:ne
     xt
52   fort=34to63
53   pokeq*64+t,(asc(mid$(b$,(t-34)*2+1
     ,1))-65)*16+asc(mid$(b$,(t-34)*2+2
     ,1))-65
54   next:return
55   print"[HOME]"tab(12)"[CTRL-9][CTRL
      1][2xSPACE]penalties:"str$(z)"[2x
     SPACE]":
56   print"[HOME][CRSR-DOWN][12xCRSR-RI
     GHT][CTRL-9][16xSPACE]"
57   print"[HOME][2xCRSR-DOWN]"tab(12)"
     [CTRL-9][2xSPACE]gestopt[2xSPACE]:
     "str$(gs)"[2xSPACE]":return
58   dataaaaaaaaaaaaaaaaaaaaabeaaaaffaa
     aahnaaaappaaakppkacklokicikkcicikk
     ciap
59   datakkpaapffpaabffeaadebmaadmdmaad
     mdmaaciciaaciciaabebeaabebeaab
60   dataaaaaaaaaaaaaaaaaaaaabeaaaaffaa
     aahnaaaappaaakppkacklokicikkcicikk
     cici
61   datakkcidnffhmdnffhmadebmaadmdmaad
     odoaaakakaaakakaaafafaaafafaab
62   dataaaaaaaaaaaaaaaaaaaaabeaaaaffaa
     aahnaaaappaaakppkacklokikikkcikakk
     cika
63   datakkcipaffhmpbffhmabhbpaaapapaaa
     oioiaaciciaaciciaabebeaabebeab
64   dataaaaaaaaaaaaaaaaaaaaaaaaaaabeaa
     aaffaaaahnaaaappaaakppkacklokikikk
     cika
65   datakkcikaffhmpaffhmpahbpaaapapaaa
     oioiaaciciaaafciaaafafaaaaafab
66   dataaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
     aafaaaabfeaaabpeaaadpmaaadpoiaakpk
     kack
67   datakkkacikkpaciffpadmhbpadmpapaaa
     oioiaaciciaaafciaaafafaaaaafab
68   dataaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
     aaaaaaaaaaaaaafaaaabfeaaabpgiaadpo
     kaal
69   datapokackpkpacikjpaciffeacihipapa
     pjoipaabgiaaaaafaaaaafaaaaaaab
70   dataaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
     aaaaaaaaaaaaaaaaaaaafaaaabfeaaabpg
     iaad
71   datapokaalpopaakpkpacikfeacifepipa
     pjojpapjefaaaaabaaaaaaaaaaaaab
72   dataaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
     aaaaaaaaaaaaaaaaaaaaaaaaabeaaaaffa
     aaah
73   datankaaappkiaappkiaalogiaakfhoaci
     ndoepipabepaoeaeaakeaaaaaeaaab
74   dataaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
     aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaabea
     aaaf
75   datafaaaahnaaaappiaaappkaaalokaack
     looecihnoepapjaepaabaaaaaaaaab
76   dataaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
     aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
     aaaa
77   dataaaaaabeaaaaffaaaahniafappkgjal
     okhickkjhacilnpjpadmpjpaaaabab
78   dataaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
     aafaaaabfeaaabpeaaadpmaaadpoiaakpk
     kack
79   datakkkacikloapafhmapahbeaaapipaaa
     cioiaaafciaaafcfaaaaafaaaaaaab
80   dataaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
     aafaaaabfeaaabpeaaadpmaaadpoaaakpk
     iack
81   datakpiaoikpaapaffaaaapbeaaadipaaa
     cjoiaaafcfaaaaafaaaaaaaaaaaaab
82   dataaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
     abeaaaaffaaaahnaaaappaaaappiaaalok
     aack
83   dataloaaoilneapaffpjaafapjaahoebaa
     doeaaaaaeaaaaaaaaaaaaaaaaaaaab
84   dataaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
     abeaaaaffaaaahnaaaappaaaappiaaalok
     aack
85   datakkaaoipjpjpapfpjaafoebaadoeaaa
     aaeaaaaaaaaaaaaaaaaaaaaaaaaaab
86   dataaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
     abeaaaaffaaaahnaaaappaaaappiaaalok
     aack
87   datakkajolojpjpdnfpbaafoeaaadoeaaa
     aaeaaaaaaaaaaaaaaaaaaaaaaaaaab
88   dataaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
     aafaaaabfeaaabpeaaadpmaaadpoiaakpk
     kapk
89   datakloapiklmaaaffeaaahbpaaapapaaa
```

```
     oioiaaciciaaafciaaafafaaaaafab
90   dataaaaaaaaaaaaaaaaaaaaaaaaafaaa
     abfeaaabpeaapdpmaapdpoiackppkackkp
     kaai
91   datakkiaaaffeaaahbpaaapapaaaoipaaa
     ciciaaafciaaafafaaaaafaaaaaaab
92   dataaaaaaaaaaaaaaaaaaaaaaaaafaaa
     pbhmaapbpmaacloiaaclpkaackpkiaakkk
     iaaa
93   datakkaaaaffeaaahbpaaapipaaacioiaa
     afciaaafafaaaaafaaaaaaaaaaaab
94   dataaaaaaaaaaaadmaaapdmaaapakaaa
     cjkeaacjoiaaaloiaaaloiaaacpkaaaakk
     aaaa
95   datakkaaaaffeaaahbpjaapidjaacjabaa
     afaaaaaaaaaaaaaaaaaaaaaaaaab
96   dataaaaaapaaaaapaaaapciaaapcjaaa
     cjkeaacjkeaaaloiaaaloiaaacpkaaaakk
     eaaa
97   datakjhjaaffhjaahoabaadkeaaaabeaaa
     aaaaaaaaaaaaaaaaaaaaaaaaaaab
98   dataaaaaaaaaaaaabaaaaafeaaaafeaa
     aapmaaakkkiaaokkmaaokkmaaokkmaaokk
     maao
99   datakkmaanffmaabffaaabefaaadmpaaad
     mpaaacikaaacikaaabefaaaaaaakk
100  dataaaaaaaaaaaaabaaaaafeaaaafeaa
     aapmaaakkkiaaokkpaaokkdmaokkpaaokk
     maao
101  datakkaaabffaaabffaaadmfaaabepaaab
     ekaaabekaaaaafaaaaaaaaaaaaakk
102  dataaaaaaaaaaaaabaaaaafeaaaafeaa
     aapmaaakkkiadokkpadckkdmdckkamdckk
     amdc
103  datakkamabffaaabffaaadmfaaacipaaab
     epaaabekaaaaakaaaaafaaaaaaakk
104  dataaaaaaaaaaaaabaaaaafeaaaafeaa
     aapmaaakkkiadokkpapckkdamckkdamckk
     damc
105  datakkdaabffaaabffaaadmfaaadmpaaac
     ipaaabekaaaaakaaaaafaaaaafaakk
106  dataaaaaaaaaaaaabaaaaafeaaaafeaa
     aapmaaakkkiadokkmapckkmadokkmaaokk
     maac
```



```
107  datakkmaabffaaabffaaabepaaadmfaaac
     ifaaacifaaabeaaaaaaaaaaaaakk
108  dataaaaaaaaaaaaabaaaaafeaaaafeaa
     aapmaaakkkiadokkpapckkdamckkdamckk
     damc
109  datakkdaabffaaabffaaabepaaadmkaaad
     mfaaacifaaaciaaaabeaaaaaaakk
110  dataaaaaaaaaaaaabaaaaafeaaaafeaa
     aapmaaakkkiadokkpadckkdmdckkamdckk
     amdc
111  datakkamabffaaabffaaabepaaadmpaaad
     mkaaadmfaaaciaaaabeaaaabeaaakk
112  dataaaaaaaaaaaaapbdmadmfepapafedm
     dmpmpaakkkiaackkaaackkaaackkaaackk
     aaac
113  datakkaaabffaaabffaaabefaaadmpaaad
     mpaaacikaaacikaaabefaaaaaaakk
114  dataaaaaaaaaaaaaaaaaaaaaaaaaaaaa
     aapaaaabpiaaadpmaaahpoaaahpoaaahpo
     aaah
115  datapoaaadpmaaabpiaaaapaaaaaaaaaa
     aaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

** EINDE LISTING penalty **

regel 1	134	regel 30	32	regel 59	166	regel 88	118
regel 2	155	regel 31	138	regel 60	203	regel 89	188
regel 3	24	regel 32	27	regel 61	200	regel 90	185
regel 4	10	regel 33	113	regel 62	211	regel 91	118
regel 5	61	regel 34	195	regel 63	213	regel 92	182
regel 6	13	regel 35	194	regel 64	171	regel 93	93
regel 7	19	regel 36	14	regel 65	211	regel 94	184
regel 8	64	regel 37	36	regel 66	105	regel 95	54
regel 9	64	regel 38	232	regel 67	223	regel 96	191
regel 10	67	regel 39	241	regel 68	41	regel 97	41
regel 11	166	regel 40	214	regel 69	236	regel 98	225
regel 12	194	regel 41	237	regel 70	247	regel 99	153
regel 13	62	regel 42	187	regel 71	6	regel 100	234
regel 14	145	regel 43	133	regel 72	221	regel 101	72
regel 15	113	regel 44	205	regel 73	16	regel 102	198
regel 16	92	regel 45	242	regel 74	209	regel 103	104
regel 17	123	regel 46	111	regel 75	236	regel 104	207
regel 18	123	regel 47	217	regel 76	199	regel 105	110
regel 19	199	regel 48	172	regel 77	255	regel 106	222
regel 20	74	regel 49	237	regel 78	105	regel 107	84
regel 21	75	regel 50	121	regel 79	179	regel 108	207
regel 22	173	regel 51	121	regel 80	95	regel 109	95
regel 23	213	regel 52	251	regel 81	137	regel 110	198
regel 24	64	regel 53	133	regel 82	93	regel 111	124
regel 25	20	regel 54	74	regel 83	140	regel 112	237
regel 26	148	regel 55	209	regel 84	93	regel 113	117
regel 27	169	regel 56	111	regel 85	134	regel 114	127
regel 28	243	regel 57	178	regel 86	93	regel 115	30
regel 29	59	regel 58	208	regel 87	137		

PRINT-OUT C 16 met Ufo en Pearldiver

Pearldiver c16

Dit is weer zo'n listing die we gekregen hebben naar aanleiding van onze prijsvraag. Het programma is ons opgevallen door de vele mogelijkheden die de programmeur in zo'n relatief kort programma heeft gekregen. Een uitgebreide uitleg bij het programma is niet nodig, alles wijst zichzelf. De bedoeling is zo veel mogelijk parels te verzamelen. Wie de meeste heeft heeft gewonnen. De maker van dit programma is: Theo Wiersma uit Zwijndrecht.

```
1    rem (c) 1987 cbm software productions
2    rem          zwijndrecht,holland
3    rem
4    rem    programmeur  : theo wiersma
5    rem    gemaakt op   : 19-04-1987
6    rem    voor         : commodore info
7    rem    computer     : c16 - plus/4
8    rem
10   rem
11   rem    (c) copyright 1987 (c)
12   rem
14   rem
20   poke52,59:poke56,59:clr:hi=0:vol8:zl=0
30   deffne(zl)=3073+(40*x2)+x1
40   poke1176,44:fora=0to64*8:poke15360+a,peek(53248+a):next:ch=0
50   x=1525:fora=0to239:readd:pokex+a,d:next
60   :
70   fora=0to151:readd:poke15872+a,d:next
80   :
90   poke65287,peek(65287)or16:poke65298,peek(65298)and251:poke65299,60
100  poke65303,113:color3,7,3:color0,1:color4,1:c$=chr$(17)+chr$(157)+chr$(157)
110  fora=0to7:forb=1to4:readd:d$(a)=d$(a)+chr$(d):nextb,a
120  fora=0to7:d$(a)=left$(d$(a),2)+c$+right$(d$(a),2):nexta:goto710
130  le=0:dn=0:sc=0:li=3
140  le=le+1:restore1090+(40*le):color1,16,1:print"[SHIFT-CLR]":pl=0
150  fora=0to10:readd$:print"[SPACE]";:forb=1to19:c=val(mid$(d$,b,1))
160  ifc=3thencolor1,13,5
170  ifc=5thencolor1,16,2
180  ifc>5thencolor1,16,6
190  printd$(c);:print"[CRSR-UP]";:nextb:print:print:nexta
200  x1=3:x2=3:char,1,23,"[CTRL-7]score.":char,15,23,"level.":printle
210  char,26,23,"high.":printhi:char,1,0,"lives.":char,29,0,"pearls.":ht=0
220  print"[HOME]"spc(7)li:ifzl<>0thenforq=1to4000:next:goto710
230  j=joy(2):ifj=3thengosub510
240  ifj=7thengosub560
250  ifj=5thengosub610
260  ifj=1thengosub660
270  dn=dn+1:color1,15,5:char,x1,x2,"":printd$(dn)
280  ifpl=20then420
290  ifdn=2thendn=0
300  char,7,23,"[CTRL-7]":printsc:sys1756
310  x=fne(zl)+80:ifpeek(x)=66thenht=ht+1:ifht=2then350
320  char,36,0,"":printpl
330  ifint(rnd(1)*6)+1=6thengosub440
340  goto230
350  char,x1,x2,"[COM-1]":printd$(0):x1=x1+2:sys1756:gosub410:li=li-1:ifli<0then370
360  le=le-1:goto140
370  char,15,12,"[CTRL-2][SPACE]game[SPACE]over[SPACE]":getk$:ifk$<>""then370
380  forq=1to5000:getk$:ifk$<>""thenq=5000
390  nextq:ifsc>hithenhi=sc
400  goto710
410  fora=8to0step-1:sound1,1,6:sound3,a,6:vola:next:forq=1to200:next:vol8:return
420  gosub500:ifle>2thenle=0:li=li+1
430  dn=0:goto140
440  bs=int(rnd(1)*17)+1:bs=bs*2+3833:ifpeek(bs)<>32then460
450  pokebs,64
460  return
470  poke65287,peek(65287)and239:poke65298,peek(65298)or4:poke65299,208
480  printerr$(er),el:end
490  fora=1to1001step250:sound2,a,2:next:return
500  restore1250:fora=1to20:readd1:readd2:sound1,d1,d2:nexta:return
510  x=fne(zl)+2:ifpeek(x)=32then550
520  ifpeek(x)=69thensc=sc+100:pl=pl+1:gosub490:goto550
530  ifpeek(x)=79thensc=sc+10:sound2,1008,4:goto550
540  return
550  char,x1,x2,"[COM-1]":printd$(0):x1=x1+2:ht=0:return
560  x=fne(zl)-2:ifpeek(x)=32then600
570  ifpeek(x)=69thensc=sc+100:pl=pl+1:gosub490:goto600
580  ifpeek(x)=79thensc=sc+10:sound2,1008,4:goto600
590  return
600  char,x1,x2,"[COM-1]":printd$(0):x1=x1-2:ht=0:return
610  x=fne(zl)+80:ifpeek(x)=32then650
620  ifpeek(x)=69thensc=sc+100:pl=pl+1:gosub490:goto650
630  ifpeek(x)=79thensc=sc+10:sound2,1008,4:goto650
640  return
650  char,x1,x2,"[COM-1]":printd$(0):x2=x2+2:ht=0:return
660  x=fne(zl)-80:ifpeek(x)=32then700
670  ifpeek(x)=69thensc=sc+100:pl=pl+1:gosub490:goto700
680  ifpeek(x)=79thensc=sc+10:sound2,1008,4:goto700
690  return
700  char,x1,x2,"[COM-1]":printd$(0):x2=x2-2:ht=0:return
710  print"[COM-6][SHIFT-CLR]":char,10,4,"RRR[SPACE]RRR[2xSPACE]R[2xSPACE]RRR[SPACE]R[2xSPACE]"
720  char,10,5,"R[SPACE]R[SPACE]R[3xSPA
```

```
     CE]R[SPACE]R[SPACE]R[SPACE]R[SPACE
     ]R"
730  char,10,6,"RRR[SPACE]RRR[SPACE]RRR
     [SPACE]RRR[SPACE]R"
740  char,10,7,"R[3xSPACE]R[3xSPACE]R[S
     PACE]R[SPACE]RR[2xSPACE]R"
750  char,10,8,"R[3xSPACE]RRR[SPACE]R[S
     PACE]R[SPACE]R[SPACE]R[SPACE]RRR"
760  char,10,11,"RR[3xSPACE]R[2xSPACE]R
     [SPACE]R[SPACE]RRR[SPACE]RRR"
770  char,10,12,"R[SPACE]R[2xSPACE]R[2x
     SPACE]R[SPACE]R[SPACE]R[3xSPACE]R[
     SPACE]R"
780  char,10,13,"R[SPACE]R[2xSPACE]R[2x
     SPACE]R[SPACE]R[SPACE]RRR[SPACE]RR
     R"
790  char,10,14,"R[SPACE]R[2xSPACE]R[2x
     SPACE]R[SPACE]R[SPACE]R[3xSPACE]RR
     "
800  char,10,15,"RR[3xSPACE]R[3xSPACE]R
     [2xSPACE]RRR[SPACE]R[SPACE]R"
810  print"[HOME][CRSR-DOWN][CTRL-2](c)
     [2xSPACE]1987[SPACE]by[SPACE]cbm[S
     PACE]software[SPACE]prod[SPACE]hol
     land":getk$:ifk$<>""then810
820  fora=1to250:sound1,(int(rnd(1)*101
     5)+1),3:gosub440:sys1525
830  getk$:ifk$<>""thena=250:nexta:gosu
     b500:zl=0:goto130
840  nexta:forq=1to1000:nextq:gosub500:
     zl=zl+1:ifzl=4thenzl=0:goto710
850  le=zl-1:goto140
860  data160,0,162,0,189,161,12,201,64,
     208,52,189,121,12,201,32
870  data208,45,232,189,121,12,201,32,2
     40,4,202,76,52,6,202,24
880  data169,64,157,121,12,105,1,157,16
     1,12,105,1,232,157,121,12
890  data105,1,157,161,12,169,32,202,15
     7,201,12,232,157,201,12,232
900  data224,38,240,3,76,249,5,200,192,
     19,240,81,24,173,250,5
910  data105,40,141,250,5,141,29,6,141,
     40,6,144,9,238,251,5
920  data238,30,6,238,41,6,24,173,1,6,1
     05,40,141,1,6,141
930  data24,6,141,35,6,141,9,6,144,12,2
     38,2,6,238,25,6
940  data238,36,6,238,10,6,24,173,46,6,
     105,40,141,46,6,141
950  data50,6,144,6,238,47,6,238,51,6,7
     6,247,5,160,0,162
960  data12,185,216,6,141,250,5,142,251
     ,5,141,29,6,142,30,6
970  data141,40,6,142,41,6,200,185,216,
     6,141,1,6,142,2,6
980  data141,24,6,142,25,6,141,35,6,142
     ,36,6,141,9,6,142
990  data10,6,200,185,216,6,141,46,6,14
     2,47,6,141,50,6,142
1000 data51,6,96,161,121,201,0,32,245,5
     ,32,245,5,96,0,0
1010 data0,10,36,32,144,152,136,128,128
     ,128,128,144,32,20,5,0
1020 data0,144,36,4,5,1,1,4,1,1,5,5,4,2
     0,80,0
1030 data0,0,2,10,10,47,46,47,0,0,128,1
     60,224,252,252,252
1040 data47,47,47,47,11,3,0,0,252,252,2
     52,252,240,192,0,0
1050 data143,188,176,176,191,62,11,62,2
     40,60,12,12,252,240,176,252
1060 data223,199,205,207,12,60,243,243,
     247,211,115,243,48,240,192,192
1070 data223,199,205,207,12,15,3,3,247,
     211,115,243,48,60,207,207
1080 data240,48,48,15,3,51,63,3,204,240
     ,243,195,15,12,48,240
1090 data11,43,47,175,191,191,47,47,224
     ,232,248,250,254,254,248,248
1100 data255,255,255,255,255,255,255,0
1110 data32,32,32,32,104,105,106,107,10
     4,105,108,109,100,101,102,103
1120 data96,98,97,99,110,111,110,111,11
     2,113,112,113,114,114,114,114
1130 data"77777777777777777777","7053555
     566444566007","7603444466033566337
     "
1140 data"7305550065545506507","7540434
     064454006507","7545530066404306607
     "
1150 data"7443500063544506307","7355433
     035504503557","7545455044404300447
     "
1160 data"7355455465455555547","7777777
     777777777777"
1170 data"7777777777777777777","7036005
     006630500337","7546505003350500007
     "
1180 data"7505005000046565007","7050555
     306666560557","7050555306666360557
     "
1190 data"7666655006536460537","7353555
     006556660557","7505444056553330557
     "
1200 data"7444333043444444447","7777777
     777777777777"
1210 data"7777777777777777777","7055553
     633663363537","7444445605635005557
     "
1220 data"7666664300330560007","7533355
     600440060007","7500003600000060007
     "
1230 data"7533303600000060067","7000005
     600000060067","7444445630000060067
     "
1240 data"7000005554444464437","7777777
     777777777777"
1250 data596,12,653,4,685,4,704,4,739,4
     ,643,12,685,4,704,4,739,4,770,12
1260 data596,12,643,4,685,4,704,4,739,1
     2,704,12,685,4,643,4,596,4,596,12
```

** EINDE LISTING pearl diver **

```
regel 1    147        regel 70    211
regel 2    9          regel 80    58
regel 3    143        regel 90    234
regel 4    98         regel 100   96
regel 5    99         regel 110   141
regel 6    224        regel 120   91
regel 7    182        regel 130   135
regel 8    143        regel 140   127
regel 9    168        regel 150   165
regel 10   143        regel 160   99
regel 11   73         regel 170   101
regel 12   143        regel 180   104
regel 13   143        regel 190   84
regel 14   143        regel 200   15
regel 20   0          regel 210   69
regel 30   163        regel 220   101
regel 40   24         regel 230   12
regel 50   60         regel 240   141
regel 60   58         regel 250   135
```

regel 260	136
regel 270	9
regel 280	120
regel 290	28
regel 300	119
regel 310	32
regel 320	176
regel 330	63
regel 340	30
regel 350	73
regel 360	8
regel 370	51
regel 380	179
regel 390	240
regel 400	33
regel 410	39
regel 420	102
regel 430	204
regel 440	95
regel 450	194
regel 460	142
regel 470	241
regel 480	203
regel 490	224
regel 500	243
regel 510	52
regel 520	151
regel 530	160
regel 540	142
regel 550	82
regel 560	49
regel 570	147
regel 580	156
regel 590	142
regel 600	83
regel 610	107
regel 620	152
regel 630	161
regel 640	142
regel 650	84
regel 660	104
regel 670	148
regel 680	157
regel 690	142
regel 700	85
regel 710	134
regel 720	206
regel 730	233
regel 740	254
regel 750	71
regel 760	113
regel 770	252
regel 780	115
regel 790	254
regel 800	209
regel 810	232
regel 820	150
regel 830	143
regel 840	110
regel 850	29
regel 860	211
regel 870	164
regel 880	6
regel 890	99
regel 900	136
regel 910	22
regel 920	122
regel 930	39
regel 940	238
regel 950	151
regel 960	20
regel 970	116
regel 980	131
regel 990	227
regel 1000	74
regel 1010	114
regel 1020	66
regel 1030	160
regel 1040	174
regel 1050	129
regel 1060	23
regel 1070	21
regel 1080	79
regel 1090	202
regel 1100	43
regel 1110	173
regel 1120	205
regel 1130	127
regel 1140	43
regel 1150	38
regel 1160	58
regel 1170	73
regel 1180	48
regel 1190	77
regel 1200	38
regel 1210	127
regel 1220	247
regel 1230	233
regel 1240	30
regel 1250	155
regel 1260	208

U.F.O. c16

De naam verraadt eigenlijk al waar dit programma over gaat. Een aantal ongeïdentificeerde vliegende objecten is zoek geraakt. Aan u de taak deze op te sporen. Het is niet zo makkelijk als het lijkt want de speurtocht strekt zich uit tot in een uitgebreid grottenstelsel. Het geheel is een creatie van Peter Boersma uit Stadskanaal.

```
1       rem ---------- u.f.o. ------------
        ---
10      color0,1:color1,2,1:color4,1:scncl
        r
20      print"[SHIFT-CLR][COM-6][5xCRSR-D
        OWN][5xCRSR-RIGHT]op[SPACE]zoek[SP
        ACE]naar[SPACE]u.f.o."
30      print"[3xCRSR-DOWN][SPACE]opdracht
        :[SPACE]zoek[SPACE]de[SPACE]tien[S
        PACE]ufo's[SPACE]":print"[CRSR-DOW
        N][3xSPACE]en[SPACE]ga[SPACE]terug
        [SPACE]naar[SPACE]start."
40      print"[2xCRSR-DOWN][CTRL-5][SPACE]
        besturing[SPACE]via[SPACE][CTRL-8]
        joystick[SPACE]1[CTRL-5][SPACE]of[
        SPACE][CTRL-8]cursor[8xSPACE][CRSR
        -DOWN]toetsen"
50      print"[3xCRSR-DOWN]"tab(4)"[COM-6]
        druk[SPACE]op[SPACE]'return'[SPACE
        ][3xCRSR-UP]":getkeya$:if a$<>chr$
        (13) then 40
60      color1,2,1:poke 65298,peek(65298)a
        nd251
70      poke 65299,peek(65299)and3or48
80      for t=832 to 849:read a:poke t,a:n
        ext t:sys 832
90      data 162,0,189,0,208,157,0,48,189,
        0,209,157,0,49,202,208,241,96
100     for t=12800to 12799+(14*8):read a
        :poke t,a:next t:printchr$(142)chr
        $(8)
110     data 0,24,60,108,255,24,36,0
120     data 0,24,60,090,255,24,8,0
130     data 0,253,253,253,0,191,191,191
140     data 253,253,253,0,191,191,191,0
150     data 0,96,241,241,0,179,183,191
160     data 191,183,179,0,241,241,96,0
170     data 128,192,48,220,129,214,170,15
        3
180     data 153,170,214,129,220,48,192,12
        8
190     data 1,2,5,36,214,153,129,214
200     data 214,129,153,214,36,5,2,1
210     data 66,40,186,108,94,245,50,76
220     data 0,24,60,058,255,24,16,0
230     data 0,0,0,24,60,090,255,24
240     data 255,66,36,24,24,36,66,255
250     gosub 260:goto 570
260     print"[HOME]";
270     print"BBBBBBGIBCCBBBGEBBBBBBBBBBBCC
        CBBBBBBBBBBB"
280     print"BBBBG[3xSPACE]ECBBBG[2xSPACE
        ]IEBBBCBBBBBCCBBBBBBBBBBB"
290     print"BBBBF[4xSPACE]EBBB[5xSPACE]E
        BCCBGBBGEEGEBBBBGEGEE"
300     print"BBBBG[5xSPACE]BBBDF[4xSPACE]
        BCCG[SPACE]EG[5xSPACE]IBBG[5xSPACE
        ]"
310     print"BBBG[6xSPACE]IBBBBB[2xSPACE]
        HBCB[10xSPACE]IBD[4xSPACE]D"
320     print"BBC[3xSPACE]DCDF[SPACE]EBBBG
        [2xSPACE]EBCBDF[3xSPACE]DCF[3xSPAC
        E]EBF[2xSPACE]DB"
330     print"BGE[2xSPACE]DBCBBDHCBB[4xSPA
        CE]EEBBG[3xSPACE]IBBF[SPACE]HBBG[S
        PACE]HGE"
340     print"G[4xSPACE]BBBBBBBBBBF[5xSPAC
        E]BF[5xSPACE]IEG[SPACE]EBF[2xSPACE
        ]BFD"
350     print"[4xSPACE]HBCBCBBBBBBBF[3xSPA
        CE]HCBDHF[3xSPACE]HF[2xSPACE]IG[2x
        SPACE]IBB"
360     print"DF[2xSPACE]IBBCCCBBBBGIBDF[S
        PACE]ICGEIBF[2xSPACE]IBF[SPACE]HF[
        3xSPACE]EB"
370     print"CG[3xSPACE]IBBCBCBBE[2xSPACE
        ]IBG[6xSPACE]IG[2xSPACE]HBE[SPACE]
        EBFD[SPACE]HB"
380     print"CF[4xSPACE]ECBBCCG[8xSPACE]H
        F[6xSPACE]CC[3xSPACE]IBG[SPACE]IB"
390     print"BBBDF[2xSPACE]ICBBG[8xSPACE]
        HCCD[4xSPACE]HCG[4xSPACE]E[3xSPACE
        ]B"
400     print"CCBG[4xSPACE]IBE[5xSPACE]HM[
        SPACE]MCCCCF[3xSPACE]ICFHD[4xSPACE
        ]DHB"
```

```
410   print"BCG[2xSPACE]HBDHCF[4xSPACE]H
      CCCCCCCCE[3xSPACE]HCCCCF[SPACE]HDB
      CC"
420   print"BBF[4xSPACE]IECE[4xSPACE]ICG
      ICCCCG[2xSPACE]HBCCCCCBCCCBBB"
430   print"CCGIDF[3xSPACE]IG[8xSPACE]IC
      CC[2xSPACE]HBBBCBEGECEIBBB"
440   print"IG[2xSPACE]IBF[6xSPACE]HF[5x
      SPACE]ECG[2xSPACE]IECBCG[6xSPACE]I
      BB"
450   print"[5xSPACE]IBFDF[2xSPACE]HBBFD
      [SPACE]HDHCF[4xSPACE]IBCF[SPACE]IB
      BF[2xSPACE]IB"
460   print"F[5xSPACE]IBBBFHBBBBG[SPACE]
      IBECEF[3xSPACE]HBCB[2xSPACE]IBBF[3
      xSPACE]"
470   print"BDFD[2xSPACE]HBBBBEIGEBF[10x
      SPACE]IBCG[2xSPACE]HBEG[SPACE]HD"
480   print"BBBG[2xSPACE]ECCEG[4xSPACE]I
      BFDBBFDHF[6xSPACE]HBBG[3xSPACE]BB"
490   print"CBG[4xSPACE]IG[3xSPACE]DF[4x
      SPACE]IBBEGIEBDFHBEG[5xSPACE]HBB"
500   print"CBF[8xSPACE]HCCCF[18xSPACE]H
      DBBBB"
510   print"CBBDFDHBBFHBCBBBFDHFDBBDBBBF
      HBBDBDBBBBB";:poke 4071,66
520   restore 540:for a=0to 3:read w,v:
      if x(v)=0then poke w,76:poke w-10
      24,17
530   next a:return
540   data 3784,0,3260,1,3598,2,3891,3
550   restore 540:for a=0to 3:read w,v:
      if w=p then x(v)=1:q=q+1
560   next a:return
570   ti$="000000":p=3610:k=65:le=3
580   get a$
590   if li=0and re=0and q=10and p=36
      10then goto 1450
600   if a$<>"" then poke p,32:poke p-10
      24,113
610   if a$="d" or a$="[CRSR-LEFT]" then
       p=p-1:dp=1
620   if a$="6" or a$="[CRSR-RIGHT]" the
      n p=p+1:dp=-1
630   if a$="5" or a$="[CRSR-UP]" then p
      =p-40:dp=40
640   if a$="r" or a$="[CRSR-DOWN]" then
       p=p+40:dp=-40
650   if p=3232 or p=3872 then gosub 106
      0:p=p-39:goto 580
660   if(p=3191 or p=3831) and re=1 then
       gosub 260:re=0:p=p+39:goto 580
670   if p=3391 or p=3791 then gosub 740
      :p=p+39:goto 580
680   if(p=3432 or p=3832) and li=1 then
       gosub 260:li=0:p=p-39:goto 580
690   p1=peek(p):if p1=76 then gosub 148
      0:goto 710
700   p1=peek(p):if p1<>32 and p1<>64 an
      d p1<>65 and p1<>75 then gosub 137
      0
710   if k=65 then k=64:else if k=64 the
      n k=75:else k=65
720   poke p,k:poke p-1024,103:goto 580
730   rem ------- scherm links ---------
740   li=1:print"[HOME]";
750   print"BBBBBBBBBBBBBBBBBBBBBBBBBBBB
      BBBBBBBBBBBB"
760   print"BBBBBBBBBBBBBBBBBBBBBBBBBBBB
      BBBBBBBBBBBB"
770   print"BBBBBBBCCCCCCBBBBBBBBBBBBBBB
      BBBBBBBBBBBB"
780   print"BBBBBBBCBBBBBCBBBBBBBBBBBBBB
      BBBBBBBBBBBB"
790   print"BBBBBBBCBBBBBBCBBBBBBGEIGEEB
      BBBBBBBBBBBB"
800   print"BBBBBBBCBBBBBBCBBBBBG[6xSPAC
      E]EBBBBBBBBBBBB"
810   print"BBBBBBBCBBBBBBCBBBBE[2xSPACE
      ]DHF[4xSPACE]IBBBBBBBBBB"
820   print"BBBBBBBCBBBBBCBBBBG[2xSPACE]
      HBBBBD[3xSPACE]EEBBBGEIBG"
830   print"BBBBBBBCCCCCCBBBBE[2xSPACE]D
      BBBBBBDF[3xSPACE]IBB[5xSPACE]"
840   print"BBBBBBBCBBBBBBBEG[3xSPACE]IB
      BBBBBBBF[3xSPACE]IG[2xSPACE]HDH"
850   print"BBBBBBBCBBBBBBG[SPACE]HF[3xS
      PACE]EIBBBBBBBDF[4xSPACE]DBBB"
860   print"BBBBBBBCBBBBBE[2xSPACE]BG[2x
      SPACE]HFDBBBBBBBBBF[3xSPACE]IBBB"
870   print"BBBBBBBCBBBBE[2xSPACE]HBF[2x
      SPACE]IBBBBBBBBBBBB[4xSPACE]IBB"
880   print"BBBBBBBCBBBBFDHBBBB[SPACE]DB
      BBBBBBBBBBBD[2xSPACE]DHBB"
890   print"BBBBBBBCBBBBBBBBBBE[SPACE]IB
      BBGEIBBBBBBBF[SPACE]IBBB"
900   print"BBBBBBBBBBBBBBBBBBF[2xSPACE]
      IBE[SPACE]D[SPACE]IBBBBBBBF[SPACE]
      IBB"
910   print"BBBBBBBBBBBBBBBBGEBBD[SPACE]H
      BFHBFDBBBBBGEG[2xSPACE]EB"
920   print"BBBBBBBBBBBBBBBBF[SPACE]IBG[S
      PACE]EBBBBBBBBBBBD[4xSPACE]IB"
930   print"BBBBBBBBBBBBBBBBBDF[3xSPACE]H
      BBBBBBBBBBBBF[5xSPACE]"
940   print"BBBBBBBBBBBBBBBBBBG[2xSPACE]H
      BBBBBBBBBBBBG[2xSPACE]HBB"
950   print"BBBBBBBBBBBBBBBBBG[3xSPACE]IB
      BBBBBBBBBBBG[2xSPACE]HBBB"
960   print"BBBBBBBBBBBBBBBBBD[4xSPACE]EB
      BBBEGIBBEG[2xSPACE]DBBBB"
970   print"BBBBBBBBBBBBBBBBBBF[4xSPACE]I
      BBG[8xSPACE]HBBBBB"
980   print"BBBBBBBBBBBBBBBBBBBF[8xSPACE]
      HBFDFHDBBBBBB"
990   print"BBBBBBBBBBBBBBBBBBBBDFHDFDFHB
      BBBBBBBBBBB";:poke 4071,66
1000  restore 1040:for a=4 to 6:read w,v
      :if x(v)=0then poke w,76:poke w-1
      024,17
1010  next a:return
1020  restore 1040:for a=4 to 6:read w,v
      :if w=p then x(v)=1:q=q+1
1030  next a:return
1040  data 3929,4,3570,5,3588,6
1050  rem ------- scherm rechts --------
1060  re=1:print"[HOME]";
1070  print"BBBBBBBBBBBBBBBBBBBBBBBBBBBB
      BBBBBBBBBBBB"
1080  print"BBBBEGIBEIBBBBBBBBBBBBBBBBBB
      BBBBBBBBBBBB"
1090  print"GIBG[6xSPACE]IBBBBBBBBBBCCCC
      CBBBBBBBBBBBBBB"
1100  print"[2xSPACE]E[2xSPACE]HF[3xSPAC
      E]HBBBBBBBBBBCBBBBCBBBBBBBBBBBBB"
1110  print"F[3xSPACE]HBB[SPACE]HDBBBBBB
      BBBBBCBBBBBCBBBBBBBBBBBB"
1120  print"BFDDBEG[SPACE]IEEGIEBBBBBBBC
      BBBBBCBBBBBBBBBBBB"
1130  print"BBBBG[9xSPACE]IBBBBBBCBBBBCB
      BBBBBBBBBBBB"
1140  print"BBBG[2xSPACE]HDFHFDF[2xSPACE
```

```
      ]EBBBBBCCCCBBBBBBBBBBBBBB"
1150  print"BBE[SPACE]HDBBBBBBG[2xSPACE]
      HBBBBBCBBBBCBBBBBBBBBBBBB"
1160  print"BBF[SPACE]IBBBBBBE[3xSPACE]E
      BBBBBCBBBBBCBBBBBBBBBBBB"
1170  print"BBBF[SPACE]IBBBBBF[4xSPACE]I
      BBBBCBBBBBCBBBBBGIBBBBB"
1180  print"BBBG[2xSPACE]IBBBBBF[3xSPACE
      ]HBBBBCBBBBBCBBBEG[SPACE]HBBBBB"
1190  print"BBG[2xSPACE]HDBBBBBBDF[SPACE
      ]IBBBBCBBBBCBBBBF[SPACE]DBBBBBB"
1200  print"BG[2xSPACE]DBBBBBBBBBE[2xSPA
      CE]BBBBCCCCCBBBBEG[SPACE]IEBBBBB"
1210  print"BD[2xSPACE]IBBBBBBBBBF[2xSPA
      CE]EBBBBGEGIBEBG[5xSPACE]IBBBB"
1220  print"BBF[2xSPACE]EIBBBBBBBBF[3xSP
      ACE]IBG[9xSPACE]HDF[2xSPACE]IBBB"
1230  print"BBBD[SPACE]HDBBBBBBBBBDF[5xS
      PACE]HFDDFHFHBBG[2xSPACE]DBBB"
1240  print"BBBG[SPACE]IBBGEGIEBBBBBDFDD
      HBBBBBBBBBG[2xSPACE]HBBBB"
1250  print"BBG[10xSPACE]IBBBBBBBBBBBBBB
      BBBE[2xSPACE]DBBBBB"
1260  print"[4xSPACE]HBDF[SPACE]HDF[2xSP
      ACE]EBBBBBBBBBBBGEGIB[2xSPACE]HBBB
      BBB"
1270  print"FDFHBBBBDBBBD[SPACE]HBBBBBBB
      BBBG[4xSPACE]E[2xSPACE]IBBBBBB"
1280  print"BBBBBBBBBBBBG[SPACE]IBBGEIBB
      EBG[8xSPACE]HBBBBBB"
1290  print"BBBBBBBBBBBBF[7xSPACE]EG[3xS
      PACE]HBBBFDDHBBBBBBB"
1300  print"BBBBBBBBBBBBBBFDBBF[4xSPACE]
      HDBBBBBBBBBBBBBBB"
1310  print"BBBBBBBBBBBBBBBBBBBBDFHBBBBBB
      BBBBBBBBBBB";:poke 4071,66
1320  restore1340:for a=7 to 9:read w,v:
      if x(v)=0then poke w,76:poke w-10
      24,17
1330  next a:return
1340  data 3976,7,3556,8,3940,9
1350  restore 1340:for a=7 to 9:read w,v
      :if w=p then x(v)=1:q=q+1
1360  next a:return
1370  v=9:poke p,74:poke p-1024,66
1380  for h=80to 0step -10
1390  v=v-1:vol v:sound 3,h,5:next h
1400  for a=1 to 1000:next a:le=le-1
1410  if le=0then goto 1450
1420  k=65:poke p+dp,k:poke p+dp-1024,10
      3
1430  poke p,p1:poke p-1024,17:p=p+dp
1440  return
1450  print"[SHIFT-CLR][CTRL-8][10xCRSR-
      DOWN]"tab(10)"score:";q*10;"%"
1460  print"[3xCRSR-DOWN]"tab(10)"tijd[S
      PACE]:";ti$:print"[2xCRSR-DOWN][2x
      SPACE]druk[SPACE]op[SPACE]'return'
      "
1470  geta$:if a$<>chr$(13) then 1470:el
      se run
1480  if li=0and re=0then gosub 550:re
      turn
1490  if li=0then gosub 1350:return
1500  gosub 1020:return
1501  rem -----------------------------
1502  rem - u.f.o.  door peter boersma -
1503  rem -                           -
1504  rem - voor commodore info 1988  -
1505  rem - i.v.m. prijsvraag jan '88  -
1506  rem -                           -
1507  rem -----------------------------
```

** EINDE LISTING ufo **

		regel 270	126	regel 1180	86	regel 2200	148	regel 3220	211
regel 1	135	regel 300	40	regel 1190	215	regel 2210	43	regel 3230	105
regel 2	254	regel 310	56	regel 1200	211	regel 2220	106	regel 3240	97
regel 3	29	regel 320	124	regel 1210	57	regel 2230	152	regel 3250	143
regel 4	34	regel 330	152	regel 1220	191	regel 2240	145	regel 3260	104
regel 5	65	regel 340	228	regel 1230	227	regel 2250	122	regel 3270	139
regel 6	23	regel 350	164	regel 1240	249	regel 2260	56	regel 3280	8
regel 7	85	regel 360	6	regel 1250	89	regel 2270	139	regel 3290	125
regel 8	52	regel 365	107	regel 1260	102	regel 2280	8	regel 3300	76
regel 9	243	regel 370	222	regel 1270	7	regel 2290	77	regel 3310	177
regel 10	54	regel 380	88	regel 1280	139	regel 2300	76	regel 3320	77
regel 20	188	regel 390	116	regel 1290	129	regel 2310	177	regel 3999	233
regel 30	57	regel 400	163	regel 1300	28	regel 2320	62	regel 4000	252
regel 40	242	regel 410	234	regel 1310	76	regel 2360	142	regel 4010	6
regel 50	181	regel 420	73	regel 1320	177	regel 2999	162	regel 4020	5
regel 60	121	regel 430	203	regel 1330	65	regel 3000	66	regel 4030	177
regel 70	106	regel 440	177	regel 1999	189	regel 3010	221	regel 4040	142
regel 80	106	regel 450	137	regel 2000	66	regel 3020	70	regel 4999	185
regel 90	184	regel 460	177	regel 2010	202	regel 3030	60	regel 5000	252
regel 100	197	regel 999	210	regel 2020	96	regel 3040	140	regel 5010	169
regel 110	228	regel 1000	66	regel 2030	9	regel 3050	209	regel 5020	178
regel 120	226	regel 1020	201	regel 2040	235	regel 3060	212	regel 5030	96
regel 130	231	regel 1030	48	regel 2050	195	regel 3070	169	regel 5040	0
regel 140	153	regel 1040	186	regel 2060	89	regel 3080	2	regel 5050	165
regel 150	182	regel 1050	146	regel 2070	41	regel 3090	34	regel 5060	96
regel 160	82	regel 1060	58	regel 2080	148	regel 3100	221	regel 5070	193
regel 170	101	regel 1070	126	regel 2090	90	regel 3110	157	regel 5990	166
regel 180	240	regel 1080	12	regel 2100	187	regel 3120	94	regel 6000	171
regel 190	125	regel 1090	142	regel 2110	201	regel 3130	208	regel 6010	231
regel 200	125	regel 1100	57	regel 2120	140	regel 3140	13	regel 6020	199
regel 210	82	regel 1110	228	regel 2130	202	regel 3150	163	regel 6030	229
regel 220	82	regel 1120	106	regel 2140	240	regel 3160	96	regel 6040	148
regel 230	27	regel 1130	188	regel 2150	106	regel 3170	33	regel 6050	237
regel 240	27	regel 1140	138	regel 2160	150	regel 3180	149	regel 6060	116
regel 250	140	regel 1150	119	regel 2170	239	regel 3190	39	regel 6070	131
regel 260	140	regel 1160	162	regel 2180	9	regel 3200	187	regel 6080	22
		regel 1170	169	regel 2190	103	regel 3210	96	regel 6090	238

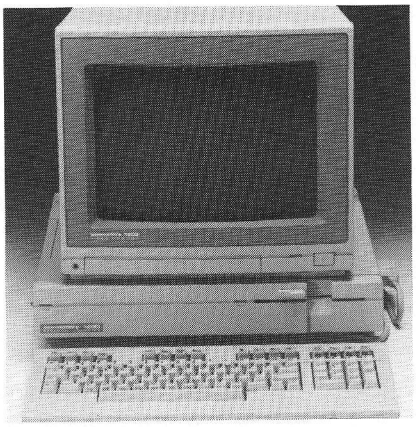
Deze maand behandelen wij in de software bespreking een softwarepakket van de in het hoge noorden gevestigde firma TEEKTRONIX. Dit pakket bestaat uit twee diskettes, waarvan ditmaal disk 1. onder de loep wordt genomen.

Interlace & Backup

Op deze diskette staan twee programma's. Het eerste programma is een Interlace editor en de tweede een supersnel backup programma. Beide programma's werken alleen in de 80 cls mode. Het backup programma is alleen bedoeld om eigen software te backuppen en niet voor illegale doeleinden!

INTERLACE-EDITOR

Interlace werkt alleen in de 80 cls mode, zodra u geen gebruik maakt van de 80 cls mode krijgt u een melding op uw scherm. Wat is nu precies Interlace? De uitvoer van een beeld op de monitor van een computer gebeurt volgens de PAL-televisie norm. Hierbij is het scherm opgebouwd uit 625 horizontale beeldlijnen, die van links naar rechts worden geschreven. Om het beeld niet te laten flikkeren stuurt de chip 50 lijnen per seconde naar de monitor. Dit betekent dus 50 maal 625 beeldlijnen per seconde. Om en om verschijnt er een beeld dat bestaat uit oneven lijnen en daarna een uit even lijnen. Het is nu mogelijk met dit programma gebruik te maken van deze Interlace mode van de 80 cls chip. Door deze Interlace mode heeft u nu 50 i.p.v 25 regels op een scherm. Dit biedt vele mogelijkheden, denk maar eens aan die enorme lange basiclisting die u nog zo verschrikkelijk nauwkeurig had ingetikt, en nu tot uw grote ergenis een typfout bevat. Wat doet u in zo'n geval: uitzoeken dus. Waar sprong die GOTO nu naar toe? Of 'in welke regel stond dat nu weer?' Dan maar weer opnieuw listen. Ja dat heb je nu met 25 regels op een scherm, maar niet diegene die dit programma bezit. Bij het opstarten van het programma komt u direct in een menu waar u de gewenste instellingen kunt uitkiezen! Een monochroom monitor zal in het algemeen beter functioneren dan een kleurenmonitor. Op een TV werkt dit programma zeker NIET!! Vanuit dit editmenu kunt u met de volgende toetsen het beeldscherm bewerken. CURSOR-UP beweegt het beeldscherm naar boven. CURSOR-DOWN beweegt het beeldscherm naar beneden. CURSOR-RIGHT verhoogd het aantal beeldscherm lijnen. CURSOR-LEFT vermindert het aantal beeldscherm lijnen. + Beweegt het scherm naar rechts. - Beweegt het scherm naar links. Waarom de programmeurs het naar links en rechts schuiven van het scherm niet onder de cursor toetsen hebben gedaan was voor ons ook verwarrend, dus jongens eventjes aanpassen in een volgende versie. SPACE verhoogt het aantal karakters op het scherm. SHIFT-SPACE verlaagt het aantal karakters op het beeldscherm. Om dit editmenu te verlaten kunt u gewoon op functietoets F3 drukken. U zult nu kunnen werken op een scherm zoals u dat zelf heeft ingesteld. Dus vanaf nu niet meer knoeien met het 25 regels tellende beeldscherm maar gewoon 'over schakelen' naar 50 regels per scherm.



LIGHTSPEEDCOPY

Het tweede programma op deze disk is Lightspeed. Dit ultra snelle kopieerprogramma werkt met twee 1570 of 1571 diskdrives. Meer heeft u niet nodig, en zelfs geen parallelkabel, maar gewoon dat oersimpele seriële kabeltje. Wij waren wat de snelheid van dit soort programmatuur al heel wat gewend. En nu komt het met dit kopieerprogramma bent u in staat in 8 SEC een kantje te kopieëren. Een dubbelzijdige diskette wordt in 16 SEC voor u gekopieerd. Stelt u zich eens voor dat naast u een kopieerfreak zit die met een enorme opgefokte diskdrive zit te werken, zodat hij een windmolen nodig heeft om zijn diskdrive koel te houden, en u met een compleet originele 128 zonder enig gesoldeer alle snelheidsrecords breekt. Wij kunnen u garanderen dat zelfs bij dit persoon de koude rillingen hem over de rug lopen. Dit programma werkt zoals boven vermeld met twee 1570 of 1571 diskdrives. Wij bedoelen daar het volgende mee: gebruik of twee 1570 diskdrives of twee 1571 diskdrives, dus niet twee verschillende. De computerfreaks die een 128/DCR bezitten moeten we helaas teleurstellen dat het programma niet samenwerkt met een 128/DCR. Dit programma is echter alleen geschikt voor het kopiëren van hele diskettes en dus geen losse file's.

Hoe werkt het?

Nadat u het programma heeft opgestart ziet u een menu. In dit menu staan de volgende functies die gekozen kunnen worden door middel van de volgende funktietoetsen:
f1 - DIRECTORY: Hiermee kunt u zowel van de source of de destination disk een directory trekken.
f3 - BACKUP: Voordat u deze funktie gebruikt zorg dan wel dat u goed in uw stoel zit. Zodra u deze functie gekozen hebt gaat het driveledje van de source diskdrive langzaam knipperen (geen error dus). Het ledje van de destination drive is dan nog gewoon uit, dit geeft u nog even de mogelijkheid om van schijf te verwisselen als dat tenminste nodig is (anders zit u straks met twee lege schijven en dit is dus niet de bedoeling van dit programma). Ondertussen wordt u de vraag gesteld of u klaar bent, als dat het geval is

drukt u op de spatiebalk. Nu wordt er gevraagd of de detination drive gereed is. Nadat u nog een keer op de spatiebalk hebt gedrukt zult u zien en horen wat er bedoeld wordt met LIGHTSPEEDCOPY.
f5 - OPTIES: Met deze functie komt u in een submenu terecht dat later besproken zal worden.
f7 - QUIT: Voor als er genoeg van heeft of klaar bent met kopieeren. Dan nu het submenu.
f2 - VERIFY: Als deze funktie op ON staat wordt er gecontroleerd op errors. Staat hij op OFF dan kopieert hij de schijf gewoon zonder errors.
f4 - SINGLE/DOUBLE SIDED: Hiermee kunnen we kiezen of u een kant of allebei kanten van de diskette wilt kopieeren.
f6 - SOURCE: Met deze optie kunt u instellen welke van de twee diskdrives nou de source is en dus tevens welke de destination is. Als de source op 8 staat wordt de destination natuurlijk 9 en net andersom vanzelfsprekend ook.
f8 - RETURN TO MAIN MENU: Met deze functie verlaat u het submenu en keert u weer terug in het hoofdmenu.
U zult reeds wel gemerkt hebben dat wij het over errors hadden. Wat is er nu zo belangrijk aan het kunnen kopieeren van errors ? Dat zullen we u even vertellen veel software firma's maken gebruik van deze errors. Door een error op een diskette te zetten van een orgineel en deze dan ook te controleren is er als het ware een kopieer beveiliging aangebracht er zijn immers maar weinig goede en tevens snelle kopieer programma's die deze errors dan ook kopieeren.

CONCLUSIE: Wij vinden dat dit pakket de moeite van het aanschaffen waard is.

JOHAN & JOHAN.

Om maar met de deur in huis te vallen, de afleveringen van Triple 128 in Commodore Info die door u gelezen worden zijn natuurlijk veel eerder door ons behandeld. En daarom is het niet mogelijk om direct alle vragen en antwoorden in het volgende nummer te publiceren. Verder moesten wij eerst de brieven eens goed bestuderen. Bij deze bedankt voor alle post die wij hebben mogen ontvangen van u. Mogen wij de volgende keer weer zo'n zak ontvangen?

Vragen en reakties

Wij willen bij deze alle lezers oproepen om op de vragen te reageren. Wanneer u zelf ook het omschreven probleem heeft meegemaakt, en op een andere manier dan ons heeft opgelost, stuur dan een brief en leg uit hoe u het probleem heeft opgelost. Uit de opgestuurde post hebben wij het meest aantrekkelijke voor alle lezers uitgezocht. Op deze manier krijgt u misschien nog wat inspiratie op computer gebied.

P.v. Smissen uit Haarlem schreef ons dat hij wat problemen heeft met de nieuwste versie van de 128. U weet nu dat die er ook is en wel in een metalen uitvoering. Zoals wij al schreven in het vorige nummer is deze Commodore 128 niet voor de volle 100% gelijk aan zijn P.V.C voorganger. Erg is dit niet maar u moet er alleen mee leren omgaan. De schrijver van de brief vraagt of wij dat eens willen uitzoeken. Tja dat is niet onze taak en zullen wij dus ook nooit gaan doen, dat moet de software schrijver zelf maar oppakken. Het zou immers te veel tijd kosten om elk programma door te nemen en aan te passen. Heeft u dus software die niet op deze DCR draait zult u bij de leverancier van de software moeten zijn.

Uit België kregen wij een brief van **François Vandebosch**. Hij vraagt waar de snelladers zijn te verkrijgen die wij in het vorige nummer hebben besproken. Het beste kun je bij GIMA printservice terecht. Hun adres is Vossenlaan 321, 6532 BE in Nijmegen. Hun telefoonnummer is 080-560491 Ook vraagt hij of de ram uitbreiding 1750 zo'n versneller is. Ja dat is hij, zeker wanneer je veel met CP/M werkt. En nee het geheugen wordt niet langzamer. Deze cartridge kost *f* 399,= en is verkrijgbaar bij elke Commodore dealer. Zijn vragenvuur is nog niet op want hij vraagt ook nog of de 1581 diskdrive van Commodore ook met de C128 samenwerkt. Ja Francois dat doet hij zeker. Met zijn massale opslagkarakter is dit een prima aanwinst voor de verzamel en ontwikkelaars onder ons. Zo'n drive kost ongeveer *f* 600,= Voor de Power Booster van Master Genius zul je tot de INFO beurs in November moeten wachten want daar zal hij uiteindelijk te koop zijn.

Arjan Benshop uit Boxtel weet niet waar hij software voor zijn Commodore 128 moet kopen. Nu Arjan wij hebben net bericht gekregen dat er in het Noorden van het land een importeur zit die software heeft voor **LOW BUDGET** gebruikers. De sofware van **TEEKTRONIX** kost slechts *f* 35,= per diskette. Persoonlijk hebben wij al twee programma's van deze firma mogen bewonderen. Het eerste programma was een kopieer-programma's voor twee 1570 of 1571 drive's in de 80 cls mode. Het tweede een 80 cls editor zodat u met 50 regels kunt werken!! Dit programma gebruikt de nog bijna niet ontdekte INTERLACE mode van de 80 cls chip. Hun adres is: TEEKTRONIX, Gooilandlaan 81, 8443 BH Heerenveen. U kunt deze programma's uitsluitend schriftelijk bestellen door per aangetekende brief een acceptgirokaart met de *f* 35,= op te sturen. En dit is zeker geen geld voor deze twee programma's. Deze firma wil op deze manier het aantal ROOF copieen tot NUL terugbrengen. Verder vraagt Arjan of DBASE II op de 128 draait. Ja dat werkt prima maar DBASE III draait niet op de 128 in CP/M

mode hoor. DBASE II werkt in de CP/M mode veel sneller met een ram uitbreiding. Wie o wie weet het adres van de CP/M gebruikersgroep in Nederland. Stuur ons even een briefje oke!

F. van den Berghe uit Amsterdam vraagt of er een MS DOS emulator voor de 128 verkrijgbaar is en zo ja waar. Na een telefoontje met computer collectief bleek er wel zo'n pakket te bestaan. Wij zullen in een volgend nummer dit pakket eens gaan bekijken.

Ook **Anton Vriend uit Reeuwijk** heeft een vraag over de 1750 ram uitbreiding. Hij vraagt of deze ram expansie module ook is in te bouwen. Nee Anton dat is deze module niet. Het is een externe module, en voordat je een C64 cartridge in de expansiepoort kan proppen zul je eerst de module moeten verwijderen. Ook zou je bij Gima een poortexpander kunnen aanschaffen. Dan is dat weer opgelost. Moet je er alleen nog voor zorgen dat de C64 module uitschakelbaar is, anders zal de 128 bij het opstarten in de C64 mode belanden. Verder vraagt hij hulp bij het maken van een printerdriver voor GEOS en een START NL-10. Wie heeft er zo'n driver beschikbaar zodat wij Anton het adres van deze persoon kunnen doorgeven? Stuur even een briefje. Hij wil met deze printerdriver met dubbele dichtheid kunnen printen!

Roeland 't Hart uit Nes a/d Amstel heeft problemen bij het programmeren van een bestandsprogramma. Hij vraagt of wij wat routines hebben voor diverse zaken. Normaal zouden wij niet op een dergelijke vraag ingaan, dit zou een gekkenhuis worden, maar bij wijze van hoge uitzondering komt hier de hele hap. Het verzorgen van een melding op het scherm, bij het openen van een seq file, zonder dat u de gegevens kwijtraakt kunt u bijvoorbeeld op de onderstaande manier oplossen. Open kanaal 15 voor in uw programma, wanneer u nu een file gaat openen dient u direct te testen of deze file wel op disk staat. Je zou dat zo kunnen oplossen.

```
open 5,8,5,"(naam),s,r": input#15,ff:if
     ff<>0 then print"Error opening
     file!!":close5: goto( opnieuw
     invoeren van de filenaam)
```

Om een error van de printer te kunnen opvangen zou je het volgende kunnen doen. Plaats dit stukje ergens in de listing en roep het met een gosub aan voordat je data naar de printer wilt sturen.

```
10002 ti$="000000"
10003 close 4:open4,4,4:close4:if
      st=-125 or st =-128 then goto
      10004:else return
10004 printchr$(147)"Uw printer
      staat niet aan!!":sleep1:goto
      10003
10005 if ti$"000010"then end: else
      sleep1:goto 10003
```

JOHAN & JOHAN.

Dit keer in de hardware hoek van Triple 128 de bouw van een RS232 interface, plus de bouw van een zogenaamde null modem kabel om twee computers met elkaar te verbinden. Het geheel zal zelfs door personen met zeer weinig ervaring in het bouwen van computer hardware, eenvoudig na te bouwen zijn.

HET RS232 INTERFACE.

Het RS232 interface is de tegenhanger van het Centronics parallelle interface. In tegenstelling tot het centronics interface is RS232 dan ook seriële datacommunicatie. Dit houdt in dat alle bits een voor een over de kabel worden gestuurd/ontvangen. Het voordeel van deze manier van datatransport is dat er minder bedrading tussen de apparatuur hoeft te liggen. Bij een Centronics parallel interface wordt er gebruik gemaakt van acht datalijnen en bij een RS232 verbinding slechts twee.

Het grote voordeel van RS232 is dat het een van de weinige dingen op het computergebied is die is gestandariseerd. De RS232 verbinding wordt vooral gebruikt bij het aansluiten van een modem. Er zijn ook printers die van een RS232 interface zijn voorzien. De RS232 verbinding kan ook gebruikt worden om twee computers met elkaar te laten communiceren.

RS232 op de 128

Voor het RS232 interface is in de CBM128/D alleen de software aanwezig. Dit wil zeggen dat er in de kernal van de computer software aanwezig is om een RS232 verbinding te ondersteunen. Er is echter geen hardware aanwezig voor een RS232 verbinding. In deze rubriek zullen wij een oplossing geven om zelf zeer eenvoudig zo'n interface te maken. Het door ons omschreven interface zal op de userpoort moeten worden aangesloten. Via het interface en uiteraard de userpoort kunt U dan ook met RS232 gaan werken. Met ons RS232 interface zal dit geen probleem meer zijn.

Het interface

Men heeft bij Commodore ook een seriële poort op de userpoort aangebracht. U kunt hier de verschillende RS232 signalen over de datalijnen laten lopen. Het enige wat dan nog een probleem oplevert is dat de officiële RS232-signalen een nivo van 12V hebben. De op de userpoort aanwezige signalen zijn echter van een 5V nivo. Door een SN7404 te gebruiken kunnen we deze 5V aanbieden aan een 12V circuit, en andersom kunnen we door middel van een zenerdiode de 12V begrenzen tot +- 5V. Dit zijn de enige wezenlijke aanpassingen die er plaats moeten vinden. Er moet natuurlijk ook nog een officiële RS232 connector geplaatst worden om alle apparatuur en kabels aan te kunnen sluiten.

Benodigdheden

Om het interface te bouwen zijn de volgende componenten (lees onderdelen) nodig.

- **-1 userpoort connector**
- **-1 RS232 connector (sub D 25 polig male)**
- **-1 weerstand van 560 ohm**
- **-1 zener diode van 4,7V**
- **-1 IC voetje 14 pens**
- **-1 IC type SN 7404**
- **- printmateriaal (fotogevoelig of gaatjesprint)**

De componenten zullen in elke electronicawinkel verkrijgbaar zijn. De componenten kunnen op een gaatjesprint of op een zelf geëtst printje worden gemonteerd.

De bouw

In **fig. 1** is het bedradingsschema van het interface geplaatst. Dit kan gebruikt worden om de schakeling op een gaatjesprint te monteren. De in **fig. 2** afgebeelde tekening is de lay-out van het sporenplan, van de soldeerzijde gezien. Deze afbeelding kan gebruikt worden om een ontwerp van het interface te maken ten behoeve van het fotografisch maken van de print. U zal in de vakhandel nog wel een paar vellen met wrijfsymbolen moeten ophalen omdat de afbeelding niet schaal 1:1 is en uitsluitend ter illustratie dient. Als de draden , die aan de print gesoldeerd moeten worden, geplaatst zijn is het interface klaar voor gebruik. Wanneer u een modem aan wilt sturen (onder andere ons Quatro, Teletron PC en de diverse externe Discovery modems) dient u ook nog enige andere controle signalen aan te sluiten. De pennen die voor deze signalen moeten worden door-

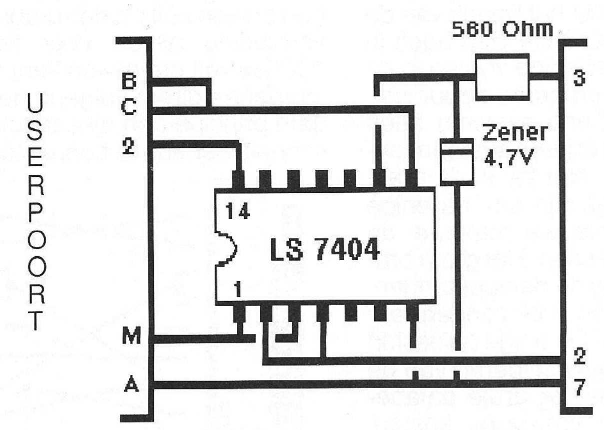


Fig. 1. bedradingsschema interface

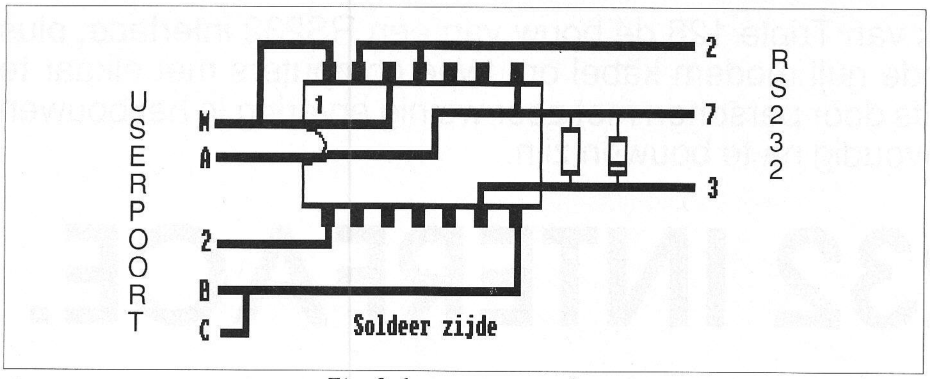


Fig. 2. lay-out sporenplan

verbonden kunt U in **tabel 1** vinden. Voor de verbinding tussen de twee connectoren kunt U het beste geen draad gebruiken met een zogenaamde harde kern. Deze is namelijk niet zo soepel en zal in het algemeen sneller op de aangesoldeerde plaatsen afbreken. Als u een verbinding tussen twee computers wilt maken is het mischien makkelijk om een stuk zogenaamde flatcable te gebruiken.

De praktijk

Wij hebben het interface uitgetest op verschillende apparaten. Ten eerste hebben wij van het interface gebruik gemaakt om te communiceren met een modem. Dit ging geheel vlekkeloos (met de extra verbindingen). Ten tweede hebben we een IBM PC met een C128 verbonden en de C128 als terminal gebruikt. Ook dit ging uitstekend. Om echter een verbinding tussen twee computers te realiseren is een speciale kabel nodig, de **null modem** kabel, maar hier komen we zo op terug

RS232 in de praktijk

Doordat de computersystemen tegenwoordig veel goedkoper zijn dan in het verleden komen ook de iets geavanceerde apparaten in het bereik van de computerhobbyist. Deze stap heeft in de praktijk nog wel enige voeten in de aarde omdat alle programmatuur welke men voor het ene systeem heeft ontwikkeld op het andere systeem onbruikbaar zal zijn. Het zal in de praktijk echter mogelijk zijn om, na enige oefening, op de nieuwe computer de software aan te passen. Het grote probleem zijn dan nog de databestanden. Want deze kunnen niet zondermeer ingelezen worden. De enige oplossing lijkt dan het opnieuw invoeren van de data. Dit zal echter bij grote databestanden vele man-/vrouwuren kosten. Maar wanneer u over een verbinding tussen de twee computers beschikt, kunt u de computer het werk laten doen. Wanneer u een RS232 verbinding tussen de twee computers maakt kunt u de data over deze kabel naar de andere computer overseinen. Deze methode is stukken sneller, en kost zeker minder energie.

Conclusie

Al met al hopen wij u hier een goedkope oplossing geboden te hebben om zelf een RS232 interface te maken. U kunt voor ongeveer 35 gulden een interface maken wat een vergelijkbare winkelwaarde van een kleine 100 gulden heeft.

De null modem kabel

Om tussen twee computers te kunnen communiceren dient u over een zogenaamde **null modem kabel** te beschikken. Deze kabel zorgt er voor dat de beide computers op de juiste manier met elkaar verbonden worden. Deze kabel is gebaseerd op de RS232 standaard. Wanneer u een andere computer op de CBM 128/D wilt gebruiken zult U ook het RS232 interface, elders in deze rubriek, moeten bouwen. Als u een verbinding tussen twee CBM 128/D's wilt maken zult u dus twee RS232 interfaces nodig hebben en een null modem kabel. Als u de verbinding alleen voor twee CBM 128/D's wilt gebruiken kunt u beter de interfaces direct volgens het null modem principe aan elkaar solderen. Dit scheelt vier sub D connectoren.

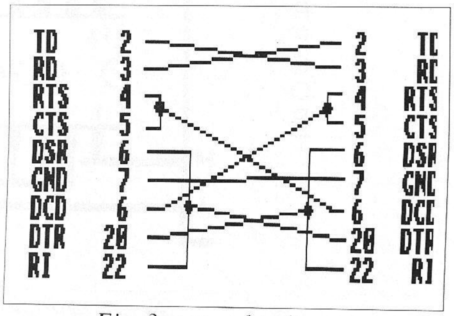
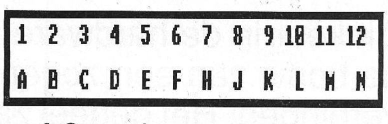
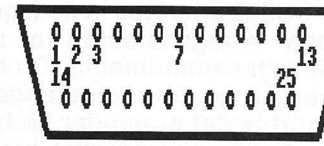


Fig. 3. penverbindingen



A Ground
B Received Data
C Received Data
D Request to send
E Data Terminal ready
F Ring indicator
H Received line signal
K Clear to send
L Data set ready
M Transmitted data
N Ground



1 Ground
2 Transmit
3 Receicve
4 Request to send
5 Clear to send
6 Data set ready
7 Ground
8 Received line signal
9 + 12V
20 Data terminal ready

De bouw van de kabel

De null modem kabel bestaat uit twee connectoren die op een bepaalde manier met elkaar verbonden moeten worden. Hoe de verschillende pennen met elkaar verbonden moeten worden vindt u in **fig. 3**. De connectoren zijn van het type sub D 25 polig female. U zult deze kabel nodig hebben om tussen PC en C128, of tussen een Amiga en een C128 te kunnen communiceren. Maar ook voor de communicatie tussen de Amiga en de PC. Zoals u ziet is voor de Amiga vrienden onder ons deze kabel ook zeer aantrekkelijk. De benodigde programmatuur zult u zelf moeten ontwikkelen. Maar wanneer blijkt dat er helemaal niemand is die het voor elkaar krijgt zullen wij in een volgend nummer de oplossing plaatsen. De naam is al bekend en zal INFO-COM zijn. Schrijf ons wanneer u behoefte hebt aan een goed terminalprogramma. Wij zullen dit programma dan voor een kleine vergoeding op disk aanbieden. Blijft er over om te vermelden dat alle handelingen door ons persoonlijk zijn getest en dat zowel het interface als de kabel perfect werken. Ga nu niet als een gek zitten knoeien met de soldeerbout maar vraag om hulp als u er niets van bakt! Die zal u graag willen helpen bij het bouwen van de hardware.

JOHAN & JOHAN.

Deze keer behandelen we een karakterset voor de 80 cls mode. De meeste personen blijken in de praktijk toch nog enige moeite te hebben met het gebruiken van eigen karakterset. Het in dit nummer behandelde programma lijkt een prima hulp. Bekijk het maar eens goed en ook u zult het vanaf nu begrijpen. In ieder geval veel plezier met ons programma.

WAAROM EEN KARAKTERSET?



Het zelf definiëren van deze karakterset is niet eenvoudig. De meeste mensen kunnen het op het 40 cls nog wel voor elkaar krijgen, vooral ook omdat er op het gebied van de 40 cls chip enorm veel lectuur op de markt is. Met de 80 cls chip is dit nog niet het geval. Maar omdat een eigen karakterset zeer mooi is plaatsen wij in dit nummer een IBM achtige karakterset. Intikken en runnen is vaak al voldoende.

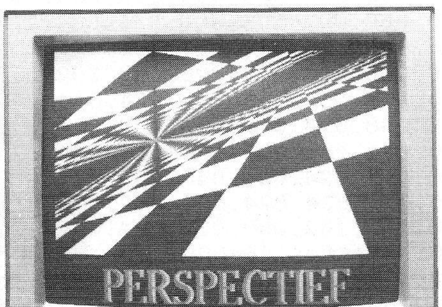
De standaard karakterset van de cbm 128 is niet echt de mooiste. In veel programma's zult u een eigen karakterset tegenkomen. Waarom komen wij zo weinig eigen karaktersets tegen? Zeker omdat bijna niemand er kaas van gegeten heeft. Daar zullen wij dan maar direct korte metten mee maken. En geloof ons, deze IBM achtige karakterset zult u vanaf nu vaker tegenkomen.

Het 80 CLS.

Het 80 cls heeft intern in de CBM 128 zijn eigen VideoRAM (VRAM). Dit is in de oudere CBM128/D 16Kb groot. In de nieuwe CBM 128DCR is hij 64Kb groot. Jazeker, elke 128/DCR heeft een VDC RAM van 64 echte kilobytes! In een volgend nummer komen we hierop terug. In dit geheugen worden alle gegevens van een beeld opgeslagen. Tevens zijn er in dit video RAM twee karaktersets aanwezig. Namelijk de gewone karakterset en een Inverse karakterset. Deze is aangebracht om ook de inverse karakters snel op het scherm te kunnen laten verschijnen.
Het geheugen is als volgt georganiseerd:
$0000-$07cf Video RAM
$0800-$0fcf Attribuut-RAM
$2000-$3fff CHARROM (kar.-set)

De karakterset

Er zijn in het VideoRAM 16 bytes gereserveerd voor elk karakter. Hiervan zijn de laatste 8 null bytes. Deze zijn om redenen van VDC interne aard aangebracht. De karakterset wordt bij het opstarten van de computer uit het CHARROM gekopieerd naar het videoram. Ook bestaat de mogelijkheid om door op de CAPS LOCK toets te drukken de karakterset te kopiëren. Zoals eerder vermeld begint de karakterset op $2000. Hiermee kunnen we dan een formule opstellen waarmee we de geheugenlocatie van elk karakter kunnen berekenen.



vb: Adres = 2*4096 + (kar)*16

Kar is hierin de ASCII waarde van het karakter waarvan u het adres wilt bepalen. *Let wel dat dit geheugenadres* in het VideoRAM zit, en er niet met de monitor in gekeken kan worden. Zoals we al vermeldden staat er behalve de gehele karakterset op $2000 in het video RAM ook nog eens de gehele karakterset in inverse achter deze normale karakterset. Als men echter niet van deze inverse karakters gebruik maakt kunt u hier ideaal eigen karakters definieren. Dit gebeurt in sommige tekstverwerkers al om bijvoorbeeld wiskundige symbolen op het scherm te laten verschijnen.

De routine

Nu is het enige probleem nog om de data van een zelf gedefinieerde karakterset in het VideoRAM te plaatsen. Hiervoor hebben wij een routine ontworpen die een karakterset van disk inleest en deze dan via bank 0 $2000 naar het VideoRAM kopieert. Met enige aanpassingen in het programma kunt u zelf ook de karakterset inladen en door middel van de routine de karakterset kopiëren. Een nadeel is dat de officiële karakterset direct wordt teruggekopieerd als men op de CAPS LOCK toets drukt. Door middel van een simpele poke kunnen we dit ver hinderen. Deze poke is

'POKE DEC("0AC5"),128

We hebben de FAST listing en een zogenaamde dataloader. De FAST listing (hierbij afgedrukt) zal voor diegenen onder u die over de FAST assembler beschikken het beste gebruikt kunnen worden, voor diegenen die niet over deze FAST assembler beschikken staat de dataloader versie ook nog afgebeeld. Verder hebben we ook nog de data van een veranderde

karakterset afgedrukt. Na het intikken zult u het resultaat zien en u zult merken dat het de moeite waard was. Vooral voor diegenen onder u, die zelf wel eens programma's schrijven die op het 80 cls werken zal het zeer zeker een opluchting zijn. Verder is er natuurlijk het voordeel dat deze nieuwe karakterset geen gedeelte van het computergeheugen afsnoept omdat de karakterdata in het VDC RAM opgeslagen is.

Het opstarten

Het opstarten van de IBM karakterset is vrij eenvoudig. U kunt het programma, de dataloader, gewoon inladen en runnen. De data's die u moest intikken voor de karakterset worden dan in de binaire vorm ingeladen. U dient dus wel eerst de karakter basic loader in te laden en te runnen. Hierdoor krijgt u een file op disk te staan onder de charrset. Met deze nieuwe file werkt het machinecode programma verder. De basic loader kunt u als backup bewaren.

De nieuwe tekens

De nieuwe tekens kunt u wel even opzoeken. Probeer shift-+ en - maar eens! Ziet u wel? Probeer ook het pondteken eens. Dat is nu een backslash '\'. Met shift + en - krijgt u de volgende tekens; {}.

Listing 1. De karakterset

```
1000 rem"*****************************************************************"
1005 rem"**      Karakterset voor de CBM 128 in de 80 cls mode !        **"
1010 rem"**      door Johan & Johan in 1988 voor COMMODORE INFO.        **"
1015 rem"*****************************************************************"
1020 :
1025 for x=4864 to 6912
1030 read a:ck=ck+a:poke x,a
1035 next x
1040 if ck<>261129 then print"Fout in data's!!":end
1045 :
1050 data 060,102,102,108,108,096,060,000,000,000,060,006,062,102,063,000
1055 data 112,048,062,051,051,051,126,000,000,000,060,102,096,102,060,000
1060 data 006,006,062,102,102,102,063,000,000,000,060,102,126,096,062,000
1065 data 028,054,048,048,120,048,048,000,000,000,062,102,102,062,006,124
1070 data 224,096,124,102,102,102,102,000,024,000,056,024,024,024,124,000
1075 data 012,000,028,012,012,012,108,056,224,096,102,108,120,108,102,000
1080 data 056,024,024,024,024,024,060,000,000,000,236,126,126,102,102,000
1085 data 000,000,252,102,102,102,102,000,000,000,060,102,102,102,060,000
1090 data 000,000,220,118,102,124,096,096,000,000,118,220,204,124,012,012
1095 data 000,000,236,118,096,096,240,000,000,000,062,096,060,006,124,000
1100 data 048,048,124,048,048,054,028,000,000,000,102,102,102,102,062,000
1105 data 000,000,102,102,102,060,024,000,000,000,230,102,126,126,108,000
1110 data 000,000,102,054,060,108,102,000,000,000,102,102,102,062,006,124
1115 data 000,000,126,076,024,050,126,000,060,048,048,048,048,048,060,000
1120 data 000,192,096,048,024,012,006,000,060,012,012,012,012,012,060,000
1125 data 024,060,102,000,000,000,000,000,000,000,000,000,000,000,255,000
1130 data 000,000,000,000,000,000,000,000,024,024,024,024,024,000,024,000
1135 data 102,102,102,000,000,000,000,000,102,102,255,102,255,102,102,000
1140 data 024,062,096,060,006,124,024,000,099,102,012,024,048,102,198,000
1145 data 060,102,060,060,103,102,059,000,024,024,048,000,000,000,000,000
1150 data 012,024,048,048,048,024,012,000,048,024,012,012,012,024,048,000
1155 data 000,102,060,255,060,102,000,000,000,024,024,126,024,024,000,000
1160 data 000,000,000,000,000,024,024,048,000,000,000,126,000,000,000,000
1165 data 000,000,000,000,000,024,024,000,000,006,012,024,048,096,192,000
1170 data 060,102,110,118,102,102,060,000,024,056,120,024,024,024,126,000
1175 data 060,102,006,028,048,102,126,000,060,102,006,060,006,102,060,000
1180 data 006,014,030,054,127,006,006,000,126,096,124,102,006,102,060,000
1185 data 060,102,096,124,102,102,060,000,126,102,006,012,024,024,024,000
1190 data 060,102,102,060,102,102,060,000,060,102,102,062,006,102,060,000
1195 data 000,000,024,000,000,024,000,000,000,000,024,000,000,024,024,048
1200 data 014,024,048,096,048,024,014,000,000,000,126,000,126,000,000,000
1205 data 112,024,012,006,012,024,112,000,060,102,006,028,024,000,024,000
1210 data 000,000,000,255,255,000,000,000,060,102,102,126,102,102,102,000
1215 data 248,108,108,124,102,102,252,000,060,102,096,096,096,102,060,000
1220 data 252,102,102,102,102,102,252,000,254,102,096,124,096,102,254,000
1225 data 254,102,096,124,096,096,240,000,060,102,096,110,102,102,060,000
1230 data 230,102,102,126,102,102,102,000,060,024,024,024,024,024,060,000
1235 data 030,012,012,012,012,108,120,000,230,108,108,120,108,108,102,000
1240 data 240,096,096,096,096,108,254,000,198,238,254,214,198,198,198,000
1245 data 230,118,118,126,110,110,102,000,060,102,102,102,102,102,060,000
1250 data 252,102,102,124,096,096,240,000,060,102,102,102,102,108,054,000
1255 data 252,102,102,124,120,108,102,000,060,102,096,060,006,102,060,000
1260 data 126,090,024,024,024,024,060,000,230,102,102,102,102,102,060,000
```

Listing 1. De karakterset (vervolg)

```
1265 data 230,102,102,102,102,060,024,000,198,198,198,214,254,238,198,000
1270 data 102,102,060,024,060,102,102,000,102,102,102,060,024,024,024,000
1275 data 126,102,012,024,048,102,126,000,024,048,048,096,048,048,024,000
1280 data 024,024,024,000,024,024,024,000,024,012,012,006,012,012,024,000
1285 data 112,219,014,000,000,000,000,000,255,127,063,031,015,007,003,001
1290 data 000,000,000,000,024,000,000,000,240,240,240,240,240,240,240,240
1295 data 000,000,000,000,255,255,255,255,255,000,000,000,000,000,000,000
1300 data 000,000,000,000,000,000,000,255,192,192,192,192,192,192,192,192
1305 data 204,204,051,051,204,204,051,051,003,003,003,003,003,003,003,003
1310 data 000,000,000,000,204,204,051,051,204,153,051,102,204,153,051,102
1315 data 003,003,003,003,003,003,003,003,024,024,024,031,031,024,024,024
1320 data 000,000,000,000,015,015,015,015,024,024,024,031,031,000,000,000
1325 data 000,000,000,248,248,024,024,024,000,000,000,000,000,000,255,255
1330 data 000,000,000,031,031,024,024,024,024,024,024,255,255,000,000,000
1335 data 000,000,000,255,255,024,024,024,024,024,024,248,248,024,024,024
1340 data 192,192,192,192,192,192,192,192,224,224,224,224,224,224,224,224
1345 data 007,007,007,007,007,007,007,007,255,255,000,000,000,000,000,000
1350 data 255,255,255,000,000,000,000,000,000,000,000,000,000,255,255,255
1355 data 001,003,006,108,120,112,096,000,000,000,000,000,240,240,240,240
1360 data 015,015,015,015,000,000,000,000,024,024,024,248,248,000,000,000
1365 data 240,240,240,240,000,000,000,000,240,240,240,240,015,015,015,015
1370 data 195,153,153,147,147,159,195,255,255,255,195,249,193,153,192,255
1375 data 143,207,193,204,204,204,129,255,255,255,195,153,159,153,195,255
1380 data 249,249,193,153,153,153,192,255,255,255,195,153,129,159,193,255
1385 data 227,201,207,207,135,207,207,255,255,255,193,153,153,193,249,131
1390 data 031,159,131,153,153,153,153,255,231,255,199,231,231,231,131,255
1395 data 243,255,227,243,243,243,147,199,031,159,153,147,135,147,153,255
1400 data 199,231,231,231,231,231,195,255,255,255,019,129,129,153,153,255
1405 data 255,255,003,153,153,153,153,255,255,255,195,153,153,153,195,255
1410 data 255,255,035,137,153,131,159,159,255,255,137,035,051,131,243,243
1415 data 255,255,019,137,159,159,015,255,255,255,193,159,195,249,131,255
1420 data 207,207,131,207,207,201,227,255,255,255,153,153,153,153,193,255
1425 data 255,255,153,153,153,195,231,255,255,255,025,153,129,129,147,255
1430 data 255,255,153,201,195,147,153,255,255,255,153,153,153,193,249,131
1435 data 255,255,129,179,231,205,129,255,195,207,207,207,207,207,195,255
1440 data 255,063,159,207,231,243,249,255,195,243,243,243,243,243,195,255
1445 data 231,195,153,255,255,255,255,255,255,255,255,255,255,255,000,255
1450 data 255,255,255,255,255,255,255,255,231,231,231,231,231,255,231,255
1455 data 153,153,153,255,255,255,255,255,153,153,000,153,000,153,153,255
1460 data 231,193,159,195,249,131,231,255,156,153,243,231,207,153,057,255
1465 data 195,153,195,195,152,153,196,255,231,231,207,255,255,255,255,255
1470 data 243,231,207,207,207,231,243,255,207,231,243,243,243,231,207,255
1475 data 255,153,195,000,195,153,255,255,255,231,231,129,231,231.,255,255
1480 data 255,255,255,255,255,231,231,207,255,255,255,129,255,255,255,255
1485 data 255,255,255,255,255,231,231,255,255,249,243,231,207,159,063,255
1490 data 195,153,145,137,153,153,195,255,231,199,135,231,231,231,129,255
1495 data 195,153,249,227,207,153,129,255,195,153,249,195,249,153,195,255
1500 data 249,241,225,201,128,249,249,255,129,159,131,153,249,153,195,255
1505 data 195,153,159,131,153,153,195,255,129,153,249,243,231,231,231,255
1510 data 195,153,153,195,153,153,195,255,195,153,153,193,249,153,195,255
1515 data 255,255,231,255,255,231,255,255,255,255,231,255,255,231,231,207
1520 data 241,231,207,159,207,231,241,255,255,255,129,255,129,255,255,255
1525 data 143,231,243,249,243,231,143,255,195,153,249,227,231,255,231,255
1530 data 255,255,255,000,000,255,255,255,195,153,153,129,153,153,153,255
1535 data 007,147,147,131,153,153,003,255,195,153,159,159,159,153,195,255
1540 data 003,153,153,153,153,153,003,255,001,153,159,131,159,153,001,255
1545 data 001,153,159,131,159,159,015,255,195,153,159,145,153,153,195,255
1550 data 025,153,153,129,153,153,153,255,195,231,231,231,231,231,195,255
1555 data 225,243,243,243,243,147,135,255,025,147,147,135,147,147,153,255
1560 data 015,159,159,159,159,147,001,255,057,017,001,041,057,057,057,255
1565 data 025,137,137,129,145,145,153,255,195,153,153,153,153,153,195,255
1570 data 003,153,153,131,159,159,015,255,195,153,153,153,153,147,201,255
1575 data 003,153,153,131,135,147,153,255,195,153,159,195,249,153,195,255
1580 data 129,165,231,231,231,231,195,255,025,153,153,153,153,153,195,255
1585 data 025,153,153,153,153,195,231,255,057,057,057,041,001,017,057,255
1590 data 153,153,195,231,195,153,153,255,153,153,153,195,231,231,231,255
1595 data 129,153,243,231,207,153,129,255,231,207,207,159,207,207,231,255
1600 data 231,231,231,255,231,231,231,255,231,243,243,249,243,243,231,255
1605 data 143,036,241,255,255,255,255,255,000,128,192,224,240,248,252,254
```

```
1610 data 255,255,255,255,231,255,255,255,015,015,015,015,015,015,015,015
1615 data 255,255,255,255,000,000,000,000,000,255,255,255,255,255,255,255
1620 data 255,255,255,255,255,255,255,000,063,063,063,063,063,063,063,063
1625 data 051,051,204,204,051,051,204,204,252,252,252,252,252,252,252,252
1630 data 255,255,255,255,051,051,204,204,051,102,204,153,051,102,204,153
1635 data 252,252,252,252,252,252,252,252,231,231,231,224,224,231,231,231
1640 data 255,255,255,255,240,240,240,240,231,231,231,224,224,255,255,255
1645 data 255,255,255,007,007,231,231,231,255,255,255,255,255,255,000,000
1650 data 255,255,255,224,224,231,231,231,231,231,231,000,000,255,255,255
1655 data 255,255,255,000,000,231,231,231,231,231,231,007,007,231,231,231
1660 data 063,063,063,063,063,063,063,063,031,031,031,031,031,031,031,031
1665 data 248,248,248,248,248,248,248,248,000,000,255,255,255,255,255,255
1670 data 000,000,000,255,255,255,255,255,255,255,255,255,255,000,000,000
1675 data 254,252,249,147,135,143,159,255,255,255,255,255,015,015,015,015
1680 data 240,240,240,240,255,255,255,255,231,231,231,007,007,255,255,255
1685 data 015,015,015,015,255,255,255,255,015,015,015,015,240,240,240,249
1690 data 000
1700 bsave"charset",b0,p4864 to p6912
```

Listing 2. Karakter loader

```
1000 rem"*********************************************************************"
1005 rem"**    Karakter loader voor de CBM 128 in de 80 cls mode!           **"
1010 rem"**     door Johan & Johan voor Commmodore INFO in 1988.            **"
1015 rem"*********************************************************************"
1020 :
1025 for x=2816 to 2936
1030 read a:ck=ck+a:poke x,a
1035 next x
1040 if ck<>13891 then print"fout in data's!!":end
1045 :
1050 data 032,081,011,169,062,141,000,255,169,000,162,019,133,218,134,219
1055 data 162,018,169,048,032,069,011,232,169,000,032,069,011,160,000,177
1060 data 218,032,067,011,200,192,008,144,246,169,000,032,067,011,136,208
1065 data 250,024,165,218,105,008,133,218,144,229,230,219,165,219,201,027
1070 data 144,221,096,162,031,142,000,214,044,000,214,016,251,141,001,214
1075 data 096,169,007,162,113,160,011,032,189,255,169,000,170,032,104,255
1080 data 169,000,162,008,168,032,186,255,169,000,170,160,019,032,213,255
1085 data 096,067,072,065,082,083,069,084,000                    1090 sys2816
```

Listing 3. Fast loader

```
1000             bnk 0
1005             org $0b00
1010 pntr        equ $da
1015 charadr     equ $1300
1020 charend     equ $1b00
1025 charset     equ $3000
1030 setlfs      equ $ffba
1035 setnam      equ $ffbd
1040 setbnk      equ $ff68
1045 load        equ $ffd5
1050 close       equ $ffc3
1055 ;
1060             jsr loadchar
1065             lda #%00111110
1070             sta $ff00
1075 main        lda #<charadr
1080             ldx #>charadr
1085             sta pntr
1090             stx pntr+1
1095             ldx #18
1100             lda #>charset
1105             jsr writereg
1110             inx
1115             lda #<charset
1120             jsr writereg
1125             ldy #0
1130 loop1       lda (pntr),y
1135             jsr writedata
1140             iny
1145             cpy #8
1150             bcc loop1
1155             lda #0
1160 loop2       jsr writedata
1165             dey
1170             bne loop2
1175             clc
1180             lda pntr
1185             adc #8
1190             sta pntr
1195             bcc loop1
1200             inc pntr+1
1205             lda pntr+1
1210             cmp #>charend
1215             bcc loop1
1220             rts
1225 ;
1230 writedata   ldx #31
1235 writereg    stx $d600
1240 writeloop   bit $d600
1245             bpl writeloop
1250             sta $d601
1255             rts
1260 ;
1265 loadchar    lda #7
1270             ldx #<name
1275             ldy #>name
1280             jsr setnam
1285             lda #0
1290             tax
1295             jsr setbnk
1300             lda #0
1305             ldx #8
1310             tay
1315             jsr setlfs
1320             lda #0
1325             tax
1330             ldy #19
1335             jsr load
1340             rts
1345 ;
1350 name        asc "charset"
```

Spring-in-het-veld-128

Het programma Spring in het veld is van Y. Rozijn. Het is een spel voor de Commodore 128 met een 40 koloms kleurenscherm. Het is een spel voor twee spelers (waar de commodore 128 er één van is), en wordt gespeeld op een bord van 10 bij 10 velden. Om de Beurt springen de spelers naar een volgend veld op het bord. In elk veld staat een cijfer, dat aangeeft over welke afstand er gesprongen moet worden. Elke keer als naar een veld wordt gesprongen, krijgt het veld een lichtere kleur. Er zijn 7 kleuren, deze staan rechts naast het spel weergegeven. De speler die naar een geel veld springt, zodat het veld wit wordt, krijgt dit veld in zijn bezit. Zijn beginletter, voor de commodore dus C, M voor mens, komt in het desbetreffende veld te staan. De zelfde speler vervolgt het spel met een sprong over een afstand ven 1. Naar een wit veld mag niet meer gesprongen worden. Het kan voorkomen dat een speler in de loop van het spel niet meer kan zetten, in dat geval gaat zijn beurt voorbij. Het betreffende veld gaat dan direct over naar de tegenstander, die het spel dan weer hervat met een sprong van 1. Als op dat moment de ander ook niet meer kan zetten is het spel afgelopen, winnaar is diegene die de meeste vakken in zijn (haar) bezit heeft. Met de joystick in poort twee kan een richting worden gekozen om te springen. Is er geen zet mogelijk dan moet je op de vuurknop drukken.

```
1    rem **********************************
2    rem **        spring in@t veld        **
3    rem **   door y.rozijn, amsterdam   **
4    rem **  c128/40kol. kleurenscherm   **
5    rem **********************************
10   gosub860:gosub980
20   do:gosub640:do:gosub420:loopuntilqx:gosub580:loop
30   rem **** test hoogste buur ****
40   f=0:ifi>=hthenifa(i-h,j)>fthenk=i-h:l=j:f=a(k,l)
50   ifi<=9-hthenifa(i+h,j)>fthenk=i+h:l=j:f=a(k,l)
60   ifj>=hthenifa(i,j-h)>fthenk=i:l=j-h:f=a(k,l)
70   ifj<=9-hthenifa(i,j+h)>fthenk=i:l=j+h:f=a(k,l)
80   return
90   rem **** test vastzetten ****
100  h=1:gosub30:iff=0ands(c)>=s(3-c)thenf=7
110  iff>0thena=6:b=f:elsea=0:b=1
120  return
130  rem **** bepaal beste zet ****
140  a=a(i,j):b=d(i,j):h=b:ifa=0thenreturn
150  ifa<6thenbegin
160  : ifb=danda>=eande>0thena=5:b=7
170  : gosub30
180  : iff=0thengosub90:elseiff=6thena=0:i=k:j=l:h=1:gosub30:b=8-f
190  bend:elsegosub90
200  a=10*a+b:ifa<zathenreturn
210  ifa>zathenp=1:elsep=p+1:ifrnd(1)*p>1thenreturn
220  za=a:zc=z:return
230  rem **** c128 zet ****
240  sprite1,1,16:d=d(x,y):e=a(x,y):za=0:zc=0:p=1:ifwthensleep2:w=0
250  ify>=dthenz=1:i=x:j=y-d:gosub130
260  ifx<=9-dthenz=3:i=x+d:j=y:gosub130
270  ify<=9-dthenz=5:i=x:j=y+d:gosub130
280  ifx>=dthenz=7:i=x-d:j=y:gosub130
290  ifzc=0thenzc=128:sleep2
300  z=zc:sprite1,0:return
310  rem **** mens zet ****
320  z$="":d=d(x,y):poke209,0
330  ifx>=dthenifa(x-d,y)>0thenz$=z$+chr$(7)
340  ifx<=9-dthenifa(x+d,y)>0thenz$=z$+chr$(3)
350  ify>=dthenifa(x,y-d)>0thenz$=z$+chr$(1)
360  ify<=9-dthenifa(x,y+d)>0thenz$=z$+chr$(5)
370  ifz$=""thenz$=chr$(128)
380  t=1:do:loopwhilejoy(2)
390  do:sprite1,1,k(t):z=joy(2):x$=chr$(z):t=t+1:ift>7thent=1
400  : gett$:ift$=f$thengosub230:x$=chr$(z):w=1
410  loopuntilinstr(z$,x$):sprite1,0:return
420  rem **** zet ****
430  c=3-c:color1,16:char1,28,3,c$(c):i=fre(1)
440  movespr1,20*x+23,20*y+49:oncgosub230,310
450  ifz=1theny=y-d:elseifz=3thenx=x+d:elseifz=5theny=y+d:elseifz=7thenx=x-d
460  a=a(x,y):d=d(x,y)
470  ifz=128thenbegin:ifqpthenqx=-1:return
480  : ifa=0thens(c)=s(c)-1
490  : a=6:c=3-c:qp=1
500  bend:elseqp=0
510  ifa<6thenbegin
520  : a=a+1:color1,k(a):gshapea$(d),20*x,20*y
530  bend:elsebegin
540  : a=0:color1,k(7):gshapea$(7+c),20*x,20*y:d(x,y)=1:s(c)=s(c)+1:c=3-c
550  bend
560  a(x,y)=a:color1,16:fori=1to2:char1,32,10+5*i,right$(str$(s(i)),2):next
570  return
580  rem **** einde ****
590  fori=3to5:char1,28,i,"[8xSPACE]":next:do:loopwhilejoy(2):color1,11
600  char1,26,22,"druk[SPACE]vuur":char1,26,23,"voor[SPACE]nog":char1,26,24,"een[SPACE]spel[SPACE]"
610  i=4+sgn(s(2)-s(1)):t=1
620  do:color1,k(t):char1,28,4,c$(i):t=t+1:ift>7thent=1
630  loopuntiljoy(2)=128:return
640  rem **** spel init ****
650  color1,k(1):graphic1,1
660  fory=0to9:forx=0to9
670  : i=int(rnd(1)*5)+1:d(x,y)=i:a(x,y)=1:gshapea$(i),20*x,20*y
680  nextx,y
```

```
690   color1,16:with2:draw1,202,0to202,1
      99
700   fork=1to7:color1,k(k):gshapea$(0),
      300,200-20*k:next
710   color1,16:fori=1to15:char1,36,i+7,
      mid$("kleurenvolgorde",i,1):next
720   draw1,284,199to284,55:draw1,202,55
      to319,55:draw1,202,170to284,170
730   draw1,202,10to319,10:char1,27,8,"a
      antal":char1,27,9,"velden"
740   char1,26,10,"in[SPACE]bezit:":gsha
      pea$(8),236,100:char1,26,15,"c-128
      :[SPACE]0"
750   gshapea$(9),236,140:char1,26,20,"m
      ens[SPACE]:[SPACE]0"
760   char1,26,22,"kijk[SPACE]goed":char
      1,27,23,"voordat":char1,26,24,"je[
      SPACE]speelt"
770   movespr1,23,49:sprite1,1,16,0,0,0,
      0:char1,26,0,"springin@tveld"
780   ifc1=0thenbegin
790   : char1,29,2,"wilt[SPACE]u":char1,
      28,3,"beginnen?":t=1
800   : do:color1,k(t):char1,30,4,"j/n":
      t=t+1:ift>7thent=1
810   : : getx$:c1=instr("jn",x$)
820   : loopuntilc1:fori=2to4:char1,28,i
      ,"[9xSPACE]":next
830   bend:elsec1=3-c1
840   s(1)=0:s(2)=0:x=0:y=0:qx=0:color1,
      16:gshapec$(0),228,34:c=c1
850   return
860   rem **** init ****
870   color0,1:color1,1:color4,1:graphic
      1,1:with1
880   circle1,12,11,7,7:paint1,12,11:sha
      pea$(0),4,3,20,19
890   fori=1to5:char1,1,1,chr$(48+i),1:s
      hapea$(i),4,3,20,19:next
900   char1,1,1,"c",1:shapea$(8),4,3,20,
      19
910   char1,1,1,"m",1:shapea$(9),4,3,20,
      19
920   scnclr:circle1,12,11,8,8:shapet$,3
      ,2,26,22:sprsavet$,1
930   scnclr:char1,0,0,"aan[SPACE]zet":s
      hapec$(0),0,0,55,7:f$=chr$(133):ke
      y1,f$
940   dima(9,9),d(9,9):fori=1to7:readk(i
      ):next:fori=1to5:readc$(i):next
950   x=rnd(-ti):return
960   data 3,5,15,6,4,8,2
970   data "c-128[SPACE]is","[SPACE]u[SP
      ACE]bent[SPACE]","c-128[SPACE]wint
      ","gelijkspel","[2xSPACE]u[SPACE]w
      int[2xSPACE]"
980   rem **** instrukties ****
990   color0,1:color4,1:color5,8:printch
      r$(11)chr$(14):graphic0,1
1000  print"[3xSPACE]S[SPACE]P[SPACE]R[S
      PACE]I[SPACE]N[SPACE]G[3xSPACE]I[S
      PACE]N[3xSPACE]@T[3xSPACE]V[SPACE]
      E[SPACE]L[SPACE]D":print:color5,2:
      rv$=chr$(18)
1010  print"Spring[SPACE]in[SPACE]'t[SPA
      CE]veld";:color5,16:print"[SPACE]w
      ordt[SPACE]gespeeld[SPACE]op[SPACE
      ]een[SPACE]bord[SPACE]van";
1020  print"[SPACE]10x10[SPACE]velden.[S
      PACE]Om[SPACE]de[SPACE]beurt[6xSPA
      CE]springen[SPACE]de[SPACE]spelers
      [SPACE](C128[SPACE]en[SPACE]Mens)[
      SPACE]";
1030  print"naar[SPACE]een[SPACE]volgend
      [SPACE]veld.[SPACE]Dat[SPACE]gaat[
      SPACE]horizontaal[2xSPACE]of[SPACE
      ]vertikaal;[SPACE]een[SPACE]cijf";
1040  print"er[SPACE]op[SPACE]elk[SPACE]
      veld[4xSPACE]geeft[SPACE]de[SPACE]
      afstand[SPACE]aan[SPACE]waarover[S
      PACE]gesprongenmoet[SPACE]word";
1050  print"en.":print:print"Als[SPACE]n
      aar[SPACE]een[SPACE]veld[SPACE]wor
      dt[SPACE]gesprongen[SPACE]dan[2xSP
      ACE]krijgt[SPACE]dat";
1060  print"[SPACE]veld[SPACE]een[SPACE]
      lichtere[SPACE]kleur.[SPACE]De[2xS
      PACE]volgorde[SPACE]van[SPACE]de[S
      PACE]kleuren[SPACE]is:":print
1070  fork=1to7:color5,k(k):printrv$"[4x
      SPACE]"chr$(29);:next:print:color5
      ,16:print
1080  print"Deze[SPACE]volgorde[SPACE]st
      aat[SPACE]ook[SPACE]rechts[SPACE]o
      p[SPACE]het[3xSPACE]speelbord.":pr
      int
1090  print"Iemand[SPACE]die[SPACE]naar[
      SPACE]een[SPACE]";:color5,8:print"
      geel[SPACE]";:color5,16:print"veld
      [SPACE]";
1100  print"springt[3xSPACE](dat[SPACE]d
      us[SPACE]";:color5,2:print"wit[SPA
      CE]";:color5,16:print"wordt)[SPACE
      ]";
1110  print"krijgt[SPACE]dat[SPACE]veld[
      SPACE]in[2xSPACE]bezit.[SPACE]Deze
      lfde[SPACE]speler[SPACE]vervolgt[S
      PACE]daarna[2xSPACE]het[SPACE]";
1120  print"spel[SPACE]met[SPACE]een[SPA
      CE]sprong[SPACE]van[SPACE]1.":prin
      t
1130  color5,15:print"[2xSPACE]"rv$"[2xS
      PACE]Druk[SPACE]op[SPACE]een[SPACE
      ]toets[SPACE]voor[SPACE]vervolg[2x
      SPACE]";:getkeyx$:print
1140  color5,16:scnclr:print"Naar[SPACE]
      een[SPACE]";:color5,2:print"wit[SP
      ACE]";:color5,16
1150  print"veld[SPACE]mag[SPACE]niet[SP
      ACE]meer[SPACE]ge-[5xSPACE]spronge
      n[SPACE]worden.[SPACE]Als[SPACE]ee
      n[SPACE]speler[SPACE]daardoor"
1160  print"niet[SPACE]meer[SPACE]kan[SP
      ACE]zetten,[SPACE]dan[SPACE]gaat[S
      PACE]diens[4xSPACE]beurt[SPACE]voo
      rbij."
1170  print"Het[SPACE]betreffende[SPACE]
      veld[SPACE]gaat[SPACE]naar[SPACE]d
      e[SPACE]tegen-stander,[SPACE]en[SP
      ACE]die[SPACE]vervolgt[SPACE]da";
1180  print"n[SPACE]het[SPACE]spel[3xSPA
      CE]met[SPACE]een[SPACE]sprong[SPAC
      E]van[SPACE]1.":print:print"Als[SP
      ACE]nu[SPACE]ook[SPACE]de[SPACE]an
      der";
1190  print"[SPACE]niet[SPACE]meer[SPACE
      ]kan[SPACE]zettendan[SPACE]is[SPAC
      E]het[SPACE]spel[SPACE]afgelopen."
      :print"Winnaar[SPACE]is[SPACE]";
1200  print"degene[SPACE]met[SPACE]de[SP
      ACE]meeste[SPACE]velden[2xSPACE]in
      [SPACE]bezit.":print:print
```

```
1210  color5,4:print"Een[SPACE]richting[
      SPACE]kiezen[SPACE]gaat[SPACE]met[
      SPACE]de[SPACE]joystick(in[SPACE]p
      ort[SPACE]2).":print
1220  print"Als[SPACE]je[SPACE]niet[SPAC
      E]kunt[SPACE]zetten[SPACE]druk[SPA
      CE]dan[SPACE]op[SPACE]de[2xSPACE]v
      uurknop.":print
1230  print"Druk[SPACE]op[SPACE]de[SPACE
      ]F1[SPACE]toets[SPACE]als[SPACE]je
      [SPACE]wil[SPACE]dat[SPACE]het[2xS
      PACE]programma[SPACE]een[SPACE]zet
      [SPACE]voor[SPACE]jou[SPACE]";
1240  print"doet.":color5,15:char1,1,24,
      "[7xSPACE]Druk[SPACE]nu[SPACE]op[S
      PACE]de[SPACE]vuurknop[8xSPACE]",1
1250  do:loopuntiljoy(2)=128:return
```

** EINDE LISTING springinveld **

regel 1	35	regel 220	57	regel 480	235	regel 740	214	regel 1000	45
regel 2	71	regel 230	176	regel 490	113	regel 750	212	regel 1010	63
regel 3	152	regel 240	182	regel 500	169	regel 760	42	regel 1020	71
regel 4	148	regel 250	213	regel 510	114	regel 770	234	regel 1030	162
regel 5	35	regel 260	187	regel 520	224	regel 780	158	regel 1040	234
regel 10	147	regel 270	190	regel 530	60	regel 790	235	regel 1050	71
regel 20	35	regel 280	218	regel 540	62	regel 800	82	regel 1060	121
regel 30	118	regel 290	16	regel 550	23	regel 810	154	regel 1070	223
regel 40	55	regel 300	61	regel 560	163	regel 820	216	regel 1080	55
regel 50	185	regel 310	5	regel 570	142	regel 830	158	regel 1090	161
regel 60	214	regel 320	222	regel 580	68	regel 840	63	regel 1100	26
regel 70	186	regel 330	73	regel 590	132	regel 850	142	regel 1110	185
regel 80	142	regel 340	42	regel 600	36	regel 860	19	regel 1120	191
regel 90	55	regel 350	68	regel 610	165	regel 870	0	regel 1130	90
regel 100	214	regel 360	45	regel 620	21	regel 880	68	regel 1140	128
regel 110	135	regel 370	137	regel 630	79	regel 890	71	regel 1150	167
regel 120	142	regel 380	209	regel 640	71	regel 900	202	regel 1160	68
regel 130	234	regel 390	148	regel 650	183	regel 910	213	regel 1170	198
regel 140	89	regel 400	195	regel 660	107	regel 920	180	regel 1180	31
regel 150	114	regel 410	199	regel 670	168	regel 930	212	regel 1190	171
regel 160	158	regel 420	210	regel 680	95	regel 940	119	regel 1200	94
regel 170	42	regel 430	209	regel 690	164	regel 950	38	regel 1210	101
regel 180	4	regel 440	112	regel 700	47	regel 960	45	regel 1220	212
regel 190	28	regel 450	59	regel 710	99	regel 970	122	regel 1230	54
regel 200	182	regel 460	4	regel 720	222	regel 980	68	regel 1240	49
regel 210	67	regel 470	251	regel 730	9	regel 990	79	regel 1250	114

Archery-128

Archery is een spel voor de commodore 128. Het is de bedoeling met een pijl en boog in de roos te schieten. Maar net zo als het in werkelijkheid is, het lijkt veel makkelijker dan het is. Allereerst moet er het aantal graden worden ingesteld. Dit is de boog die de pijl moet afleggen. Deze instelling is erg belangrijk. De roos loopt van boven naar beneden over het scherm.Op het moment dat je denkt te moeten schieten druk je op de vuurknop. Bij elke 250 punten wordt het iets moeilijker en er zijn 15 levels. De maker van dit schoons is Pieter Post uit Urk.

```
1     rem    +++++++++++++++++++++++
2     rem    +++   pieter post   +++
3     rem    +++                 +++
4     rem    +++      urk        +++
5     rem    +++++++++++++++++++++++
10    fast:vol15:printchr$(14)chr$(11)
20    fors=1to40:l$=l$+chr$(32):next
30    fors=1to5:p$=p$+chr$(192):next
40    fors=0to2:reada$(s):next:gosub1000
50    fors=4to22step2:reada,a$:color5,rn
      d(0)*14+3:char1,a,s,a$:next
60    slow:fors=25to1step-1:poke53280,s:
      gosub2000:next
70    gosub1100:fast:ifa$="*"thende=1
99    rem********sprites***************
      ***
100   graphic1,1:draw1,6,11to24,11
110   draw1,8,11to6,9:draw1,8,11to6,13
120   fors=1to4:gosub1200:next
130   graphic1,1:fors=1to4:draw1,20-s,(s
      -1)*3to20-s,22-(s-1)*3:next
140   box1,19,0,24,21,,1:s=5:gosub1200
150   movesprs,280,0:sprites,0,10
160   graphic1,1:char1,0,1,"loo":s=6
170   gosub1200:movesprs,160,20:sprites,
      1,8
180   fors=7to8:y$="":fort=1to63:reada$:
      y$=y$+chr$(dec(a$)):nextt
190   sprsavey$,s:sprites,0,10,1,1,1:mov
      esprs,25,150:nexts
199   rem********spelcyclus*************
      **
200   fori=1to15:         rem(15 levels)
210   forj=1to3:          rem(3 pogingen)
220   movespr5,180#i:p=0:au=1:gosub1220
230   gosub1300:gosub1400
240   fors=1to4:movesprs,21,35+s*15:next
250   forx=1to4:          rem(4 pijlen)
260   ifde=0thenchar1,14,24,"bepaal[SPAC
      E]hoek",1
270   color1,2:char1,2,21,"00",1
280   movesprx,43,150:h=0:slow
290   ifde=0thendountiljoy(2)=1:loop
300   do:sound1,h*700,2:ifh=90thenexit
310   h=h+3:char1,2,21,right$(str$(h),2)
      ,1
320   ifdethenloopuntilh=45:elseloopwhil
      ejoy(2)=1
330   movesprx,25,150:sprite7,0:sprite8,
      1
340   color1,1:char1,0,24,l$,1
350   ifdethengosub1610:goto400
360   char1,13,24,"vuurknop",1:ifcthench
      ar1,22,24,"of[SPACE]*",1
370   do:geta$:loopuntila$="*"orjoy(2)>1
```

```
     27
380  ifa$="*"andc=0then370
390  ifa$="*"thenc=c-1:gosub1350:gosub2
     400:gosub1600
400  movesprx,90#15:sprite7,1:sprite8,0
410  gosub2200:char1,0,24,l$,1
420  dountilrsppos(x,0)=>250:loop
430  movespr5,0#0:k=150-rsppos(5,1):q=0
440  movesprx,0#0:movesprx,270,150
450  ifabs(k)<21thengosub2100:gosub1500
     :elsegosub2300
460  movespr5,180#i:spritex,0
470  p=p+q:c%=rnd(0)*r*4
480  ifqandde=0andc%=0thenc=c+1:gosub13
     50
490  color1,2:char1,1,x*2,str$(q):color
     1,1
500  nextx:sleep1:ifdethenrun
510  fast:au=0:gosub1220:gosub1300
520  char1,21,0,"aantal[SPACE]punten:"+
     right$(str$(p),3),1
530  a=0:b=10:color1,2
540  ifp=>250thena=1:ifi<15thenchar1,33
     ,10,str$(i+1)
550  ifa=0thena=2:char1,32,10,str$(3-j)
560  char1,(40-len(a$(a)))/2,b,a$(a)
570  b=12:color1,1:slow
580  ifi=15anda=1thena=0:sleep1:goto560
590  ifj=3anda=2thena=0:sleep1:goto560
600  ifathenchar1,0,24,l$,1:else650
610  char1,16,24,"vuurknop",1
620  dountiljoy(2)>127:a%=rnd(0)*16+1
630  ifa=1thengosub2600:color4,a%:elseg
     osub2300
640  loop:fast:ifa=2thennextj:elsenexti
650  ifi=15thengosub2700:elsegosub2800
660  sleep8:fast:gosub1000:ke=1
670  gosub1100:ifa$<>"*"thenrun
680  au=0:gosub1220:graphic0,1:list-5:e
     nd
998 rem+++++++++subroutines+++++++++++
     ++
999 rem********titel*****************
     **
1000  color0,2:color4,2:graphic1,1
1010  fors=1to7:a$=mid$("archery",s,1)
1020  char1,0,1,chr$(128)+a$+chr$(128),1
1030  movesprs,0#0:movesprs,s*46-24,36
1040  gosub1200:nexts:graphic0,1:return
1099 rem*********play****************
      *
1100  au=1:gosub1220:slow
1110  do:a%=rnd(0)*14+3:b%=rnd(0)*7+1
1120  play"t8i"+chr$(64+b%):spriteb%,1,a
      %:color5,a%
1130  ifkethenchar1,6,24,"OPNIEUW=vuurkn
      op[2xSPACE]STOPPEN=*",1:goto1150
1140  char1,10,24,"SPEL=vuurknop[2xSPACE
      ]DEMO=*",1
1150  geta$:loopuntiljoy(2)>127ora$="*":
      return
1199 rem******spritesubroutines********
      *
1200  shapea$,0,1,23,21:sprsavea$,s
1210  sprites,0,1,1,1,1:return
1220  fors=1to7:sprites,au:next:return
1299 rem********opschrift*************
      *
1300  color0,16:color1,1:graphic1,1
1310  char1,0,0,l$,1:char1,0,24,l$,1
1320  ifdethenchar1,5,0,"demo",1:char1,3
      0,0,"demo",1:return
1330  char1,2,0,"level:"+right$(str$(i),
      2),1
1340  char1,11,0,"poging:"+str$(j),1
1350  char1,21,0,"c=128[SPACE]shots:"+ri
      ght$(str$(c),2),1:return
1399 rem*******speelscherm************
      *
1400  with2:color1,10
1410  draw,300,8to300,191:color1,1
1420  fors=1to4:circle1,100,45,s*5.25:ne
      xt
1430  char1,0,1,chr$(213)+p$+chr$(201)
1440  fors=1to7:char1,0,1+s,chr$(221)
1450  char1,6,1+s,chr$(221):next
1460  char1,0,9,chr$(202)+p$+chr$(203)
1470  box1,14,167,32,176:color4,1:return
1499 rem********raak*****************
      *
1500  h=24+h/(90/42):circle1,100+k,h,1
1510  r=sqr((45-h)^2+k^2):q%=r/5.25+.05
1520  ifq%<5thenq=100-q%*25
1530  ifq%thenreturn:elsegosub2500
1540  movespr6,180#5:sleep1:movespr6,0#5
1550  sleep1:movespr6,0#0:return
1599 rem*******de c=128 schiet*********
      *
1600  char1,12,24,"de[SPACE]c=128[SPACE]
      schiet",1
1610  cs=150-16.2*i:ifi>9thencs=cs+255
1620  do:loopuntilrsppos(5,1)<=cs
1630  do:loopuntilrsppos(5,1)=>cs:return
1999 rem*******sounds*****************
      *
2000  sound1,7000,s,1,0,7000/s,2:return
2100  sound1,3500,3,,,,2,500:return
2200  sound1,60000,5,,,,3:return
2300  sound1,40000,20,1,0,2000,2,3000:re
      turn
2400  sound1,3000,10,0,1000,100,2:return
2500  sound1,30000,100,2,0,600,2:return
2600  fors=1to3:fort=1to3:soundt,3000+i*
      500,10,1,i*500,100,s:nextt,s:retur
      n
2700  fors=1to3:sounds,30000,400,2,,3600
      ,2:next:return
2800  fors=1to3:sounds,8000,400,1,0,20,1
      :next:return
2999 rem*********data titels**********
      *
3000  datagame over,u bent toegelaten to
      t level,resterend aantal pogingen
3010  data15,"15[SPACE]Levels"
3020  data10,"joystick[SPACE]in[SPACE]po
      ort[SPACE]2"
3030  data7,"joystick[SPACE]up[SPACE]=[S
      PACE]hoek[SPACE]bepalen"
3040  data10,"vuurknop[SPACE]=[SPACE]zel
      f[SPACE]schieten"
3050  data17,"*[SPACE]=[SPACE]C=128[SPAC
      E]shot"
3060  data7,">=[SPACE]250[SPACE]pnt.[SPA
      CE]=[SPACE]volgend[SPACE]level"
3070  data7,"Buitenste[SPACE]cirkel[SPAC
      E]-[2xSPACE]25[SPACE]pnt."
3080  data24,"-[2xSPACE]50[SPACE]pnt.",2
      4,"-[2xSPACE]75[SPACE]pnt."
3090  data7,"Binnenste[SPACE]cirkel[SPAC
      E]-[SPACE]100[SPACE]pnt."
3999 rem********data bogen************
4000  data0,1,c0,0,1,60,0,1,30,0,1,18,0,
```

print-out print-out print-out print-out print-out

```
     1
4010 datac,0,1,6,0,1,3,0,1,3,0,1,3,0,1,
     2
4020 data0,1,2,0,1,2,0,1,3,0,1,3,0,1,3,
     0
4030 data1,6,0,1,c,0,1,18,0,1,30,0,1,60
4040 data0,1,c0,0,1,c0,0,2,60,0,4,30,0,
     8
4050 data18,0,10,c,0,20,6,0,40,3,0,80,3
4060 data1,0,3,2,0,2,4,0,2,2,0,2,1,0,3,
     0
4070 data80,3,0,40,3,0,20,6,0,10,c,0,8
4080 data18,0,4,30,0,2,60,0,1,c0
```

** EINDE LISTING archery **

regel 1	108	regel 260	106	regel 560	24	regel 1220	106	regel 2000	79
regel 2	146	regel 270	219	regel 570	100	regel 1299	7	regel 2100	201
regel 3	56	regel 280	115	regel 580	91	regel 1300	255	regel 2200	57
regel 4	149	regel 290	223	regel 590	42	regel 1310	244	regel 2300	117
regel 5	108	regel 300	1	regel 600	21	regel 1320	107	regel 2400	179
regel 10	104	regel 310	181	regel 610	141	regel 1330	202	regel 2500	137
regel 20	110	regel 320	51	regel 620	0	regel 1340	207	regel 2600	21
regel 30	126	regel 330	87	regel 630	243	regel 1350	241	regel 2700	53
regel 40	154	regel 340	167	regel 640	128	regel 1399	60	regel 2800	209
regel 50	96	regel 350	103	regel 650	79	regel 1400	43	regel 2999	240
regel 60	60	regel 360	248	regel 660	211	regel 1410	181	regel 3000	186
regel 70	9	regel 370	31	regel 670	124	regel 1420	223	regel 3010	10
regel 99	39	regel 380	37	regel 680	24	regel 1430	23	regel 3020	33
regel 100	68	regel 390	118	regel 998	164	regel 1440	108	regel 3030	154
regel 110	243	regel 400	32	regel 999	169	regel 1450	98	regel 3040	159
regel 120	213	regel 410	131	regel 1000	206	regel 1460	31	regel 3050	155
regel 130	246	regel 420	223	regel 1010	44	regel 1470	66	regel 3060	32
regel 140	172	regel 430	131	regel 1020	114	regel 1499	70	regel 3070	203
regel 150	244	regel 440	40	regel 1030	210	regel 1500	156	regel 3080	198
regel 160	36	regel 450	216	regel 1040	204	regel 1510	217	regel 3090	232
regel 170	133	regel 460	93	regel 1099	93	regel 1520	88	regel 3999	134
regel 180	170	regel 470	187	regel 1100	99	regel 1530	153	regel 4000	73
regel 190	151	regel 480	57	regel 1110	180	regel 1540	243	regel 4010	64
regel 199	92	regel 490	193	regel 1120	213	regel 1550	37	regel 4020	41
regel 200	2	regel 500	205	regel 1130	156	regel 1599	189	regel 4030	31
regel 210	41	regel 510	236	regel 1140	132	regel 1600	99	regel 4040	94
regel 220	21	regel 520	22	regel 1150	194	regel 1610	8	regel 4050	42
regel 230	221	regel 530	98	regel 1199	63	regel 1620	190	regel 4060	45
regel 240	14	regel 540	3	regel 1200	84	regel 1630	132	regel 4070	249
regel 250	164	regel 550	76	regel 1210	240	regel 1999	175	regel 4080	21N

Kleine Advertenties

Te koop C-64
Compleet met stofkap, 1530 Datarecorder, MCS 801 printer, Final cartridge, KCS Power cartridge, spelletjes en programma's. *f* 1.080,- Tel. 01821-2888 na 16.00 u.

Aangeboden Printer
Brother HR5C Matrix printer compleet met papier, lint, voeding, handleiding. *f*. 250,- Tel.053-333140.

C-16 te koop
Met 64k uitbreiding, joystick, boeken, software. *f* 600,- Tel.04929-61828.

Gezocht Diskdrive
Voor de C-64 met spoed gevraagd, tel.05270-14534

Te koop Printer voor C-64
General Electric, Letter Quality, niet gebruikt, incl. Interface kabel. *f* 350,- Tel.01879-1569.

Commodore 128 te koop
Compleet met Diskdrive, Monitor, MPS Printer. *f* 1.300,- Tel.02996-3342 na 18.00 u.

Gevraagd Originele Games
Adventure Games met handleiding en in Orginele verpakking. Tel. 05437-73354 .

Te koop C-64
Compleet met Diskdrive, 2 joysticks ,Power Cartirdge, Casseterec., Diskettes, Turbo tool, boeken. *f* 750,- Tel. 02522-10517.

Printer te koop
Juki 2200 met extra Daisywheel, Centr.interface. *f* 500,- Tel.085-233323.

Aangeboden C-64
Compleet met twee Diskdrive's, kl. monitor, Seikosha printer en div. Prijs compleet *f* 2.000,- Tel.02975-62229.

Contact gezocht
Gerrit zoekt contact met andere Amiga liefhebbers. Tel.010-4256706.

C-64 te koop
Met Diskdrive, monitor, cass.recorder, twee joysticks. Tel.010-4221279.

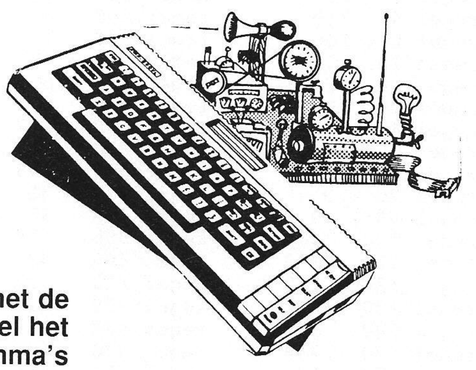
Aangeboden Originele Software
Voor informatie bellen naar 02158-1672 na 18.00 u.

Gevraagd C-64
Liefst compleet met Diskdrive, *f* 300,- Tel.020-90865.

Te koop C Plus/4
Compleet met 1551 Diskdrive, Datarec., joystick, boeken. *f* 450,- Tel.020-880164.

In onze serie Basic-artikelen zijn we inmiddels al een aantal jaren aan het werk en we hebben daardoor een groot scala basis-onderwerpen in deze succesvolle cursus aan de orde kunnen stellen.
Om alle stof die we in het verleden al of niet uitvoerig hebben behandeld nu eens gestructureerd op een rij te zetten, begint Jan Bodzinga vanaf deze aflevering met een soort alfabetische excursie langs alle standaard Basic opdrachten, functies en commando's.

Basis Basic



deel 23: Basic van A tot Z

Het is al een hele tijd geleden, dat we voorzichtig zijn begonnen met de eerste les in deze cursus. Inmiddels zijn we allemaal uiteraard wel het één en ander wijzer geworden, zeker wat betreft het maken van programma's in Basic en het werken met onze eigen computer.

Daarom leek het me leuk, om nu per Basic-opdracht een zo uitgebreid mogelijke omschrijving te gaan geven van alle standaard-opdrachten. Dat betekent, dat we naast de omschrijving van het commando zelf, ook een hele serie voorbeelden geven van de manier waarop dit bepaalde commando kan worden gebruikt.
Het feit, dat we in deze serie alleen de standaard-opdrachten van Basic behandelen heeft een diepere oorzaak. Het is in deze tijd van talloze verschillende Basic-verises helaas niet meer te doen, om alle afzonderlijke dialecten te gaan noemen, zeker waar het de aparte instructies betreft, die uitsluitend machine- of type gericht zijn. We zouden daardoor in een groot en verwarrend oerwoud van Basic-opdrachten verzeild raken, en daar heeft niemand wat aan.

Wel kun je in alle gevallen bij de redactie van dit blad aankloppen met problemen en andere lastige onderwerpen, voorzover ze je computer betreffen althans. We hebben daarvoor een hele toegewijde afdeling in het leven geroepen, en zullen daarom ook alles doen om je te kunnen helpen (Zie voorin dit blad bij de Colofon het nummer voor de listing-telefoon). Ook is het zo, dat juist de afwijkende, machine gerichte commando's in de handboeken ruimschoots aandacht krijgen, terwijl juist de heel eenvoudige functies en opdrachten als ABS, PRINT en REM heel vaak worden vergeten.

Waar het echt niet anders kan, zullen we afwijken van onze standaard-gedachte en de Commodore-64/128 beschouwen als prototype. Dit, omdat deze machine(s) nog steeds bij de oeroude basis van de echte (Commodore) Basic-familie behoren. Maar gelukkig vinden we niet erg veel heel specifieke instructies onder de standaard-vlag, zodat je rustig met je Amiga of PC kunt aanloggen. Want ook GWBasic en IBM's BASIC(A) vallen gelukkig helemaal binnen de standaard Basic.



Vooraf

Voordat we beginnen met de toch wel omvangrijke lijst van Basic-opdrachten eerst een les met de nodige algemene theorie en wetenswaardigheden. Hoewel het hier en daar niet de meest sappige stof is, ben ik er zeker van, dat in deze les het nodige te vinden is, wat zelfs de meer ervaren Basic-programmeur nog niet wist, en vanaf nu best eens zou kunnen gaan gebruiken in z'n eigen programma's.
Alle Basic-opdrachten laten we vergezeld gaan van uitleg, voorbeelden, opmerkingen en andere details, die anders gemakkelijk over het hoofd kunnen worden gezien.
Het FORMAAT waarin de Basic-opdrachten zal worden weergegeven is (hoop ik) min of meer eensluidend. Dik gedrukt het commando (of de functie) waarbij wordt aangegeven of het een Basic-opdracht of een Basicfunctie betreft. Deze laatste kunnen alleen worden gebruikt, als samen met de opdracht, tussen haakjes ook een expressie wordt meegegeven.
Voorbeelden van functies zijn SQR, LOG, EXP, ABS, LEFT$, MID$ en ga zo maar door. In het verloop van deze les zullen we meer over functies vertellen en de serie laat ons nog genoeg van deze functies tegenkomen.

Omdat voor de beginners het hiervoorgaande misschien ook niet veel anders dan abacadabra is, zullen we voor we in de komende les met de echte lijst beginnen, wat dieper ingaan op de wijze waarop we Basic in het algemeen moeten schrijven, de **Basic-syntax** zogezegd.

Syntax

De 'syntax' van de taal Basic, wil niet veel anders zeggen, dan 'manier waarop Basic moet worden beschreven' ofwel 'Basic-schrijfwijze'.

Dat dit belangrijk is, zal iedereen kunnen begrijpen, want de taal (Basic) wordt gebruikt als communicatie tussen het inwendige van de computer en de programmeur of computer-gebruiker. Door een aaneenschakeling van opdrachten - of instructies- maakt de gebruiker aan de computer bekend wat er moet gebeuren. We schrijven met een programma als het ware een brief naar de microprocessor. Het verschil tussen een brief aan je vriend of vriendin en een brief aan de computer ligt in het feit, dat ,als je in je humane brief een woord verkeerd schrijft, of een leesteken vergeet, de ontvanger zonder meer wel zal begrijpen wat je bedoelt. Jouw brief wordt door de lezer -ondanks de fouten- goed geïnterpreteerd.
De computer reageert echter heel anders op de brief, of het programma waarmee hij wordt gevoerd. Als puur logische machine is het apparaat volstrekt niet in staat ook maar iets te begrijpen van jouw fouten. Hij heeft intern een 'vertaal-routine', de Basic-interpreter, waarmee de pure Basic-taal en expressies, worden vertaald en uitgevoerd. Bij foute commando's laat het ding onmiddelijk alles in de steek, 'denkt' niet deze PINT zal wel als PRINT bedoeld zijn, maar geeft een onverbiddelijke '**Syntax Error**', ofwel schrijffout. Gelukkig reageert de computer dus nog enigszins op de fout, anders kunnen we echt lang zoeken naar iets onschuldigs als een vergeten puntkomma.
Deze Basic-syntax, die overigens veel lijkt op eenvoudig huis-tuin-en-keuken-engels, zal net als iedere andere computertaal moeten worden geleerd. Zonder die kennis is het niet te doen om een Basic programma te schrijven. Gelukkig is Basic één van de meest eenvoudig en snel te leren computertalen, en op dit moment zelfs bijna even krachtig als de wat professioneler bekende talen als bijvoorbeeld PASCAL, C, of COBOL.

Contekst

Naast de directe schrijfwijze die we foutloos in ons programma moeten toepassen komen we bij het lezen van het programma nog een ander fenomeen tegen, de 'contekst' waarin het commando wordt geplaatst. De opdracht RETURN is als zodanig helemaal goed, en niemand zal iets op de volgende regel aan kunnen merken:

```
20 RETURN
```

Als dit de eerste regel uit ons programma is, dan zitten we wel in de problemen, want de computer zal onmiddelijk reageren met een melding als 'RETURN WITHOUT GOSUB ERROR' terwijl de syntax juist is. De samenhang waarin de opdracht is gezet klopt dus niet in dit geval.
Als dit programma wat wordt uitgebreid tot :

```
20 RETURN
30 REM BEGIN HIER
50 GOSUB 20
60 PRINT "KLAAR"
```

en we typen in plaats van RUN nu de opdracht RUN 30, dan zal de foutmelding verdwenen zijn, terwijl regel 20 nog steeds de eerste regel van het programma is, en de contekst van deze regel is ineens in orde. Soortgelijke problemen kunnen we verwachten bij opdrachten als NEXT, GOTO, READ, RESTORE om nog maar te zwijgen over NEW. Wat ik hier het meest mee zeggen wil, is dat een lijst van opdrachten en functies natuurlijk niet alles zegt. Want de opdracht :



'NEW#@#@#%$$#@'

zo ingetypt als hier staat, is qua syntax geheel fout, maar zal toch feilloos door de computer worden uitgevoerd, omdat na NEW niets meer komt dan een opnieuw leeg computergeheugen, dus ook **NEW#@#@#%$$#@** bestaat niet meer. Aan de andere kant is een FOR zonder NEXT wat schrijfwijze betreft helemaal goed, maar zal uiteindelijk toch niet doen wat we verlangen, en geeft ook voor de computer aan het eind van het programma forse problemen met 'stack-behandeling' en dergelijke.
Natuurlijk zijn ook voor de Basic-notatie de nodige regels opgesteld, waarbij we in deze serie uitgaan van een standaard, geïnsprireerd op de wijze waarop Backus-Naur met notaties bezig is geweest. We nemen echter de vrijheid om alles een beetje losser en beschrijvender te nemen, zodat -naar ons idee- de opdrachten heel gemakkelijk te begrijpen en toe te passen zijn.

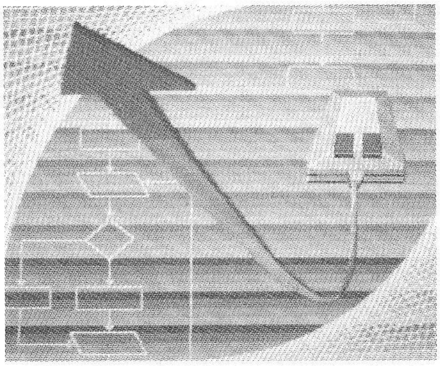
Constanten

Nog even wat over de manier waarop in de meeste gevallen Basic 'konstanten' te onderscheiden zijn van variabelen. Zeker als je per Basic-opdracht het optimale profijt uit de voorbeelden wilt trekken, is het essentieel dat dit onderscheid goed wordt begrepen. Vooral als we onverhoopt te maken krijgen met numerieke waarden waarin de letter 'E' als aanduiding voor exponent wordt gebruikt.
Numerieke en string-konstanten worden gebruikt om een waarde aan het programma mee te geven die in alle gevallen dezelfde is. Vandaar ook de naam 'konstante'. Deze konstante waarden hebben we bijvoorbeeld nodig om een lus in het programma een vast aantal keren uit te voeren. Of de waarde van 'PI', die we als konstante altijd nodig hebben als we met bepaalde wiskundige berekeningen bezig gaan. De notatie van deze konstante numerieke en alfanumeriek waarden wijkt nogal af van de standaard schrijfwijze die voor Basic-variabelen gebruikelijk is.
Waarden als **5**, **0.0**, **+45.7**, **-2.3000**, **1.44453E-10**, **1234.567** en **-3342** zijn numerieke konstanten,
terwijl **"Daag"**,**"&&$%#@"**,"   ", en **"-223323"** alfanumeriek konstanten, ofwel strings, zijn. De laatste zijn te herkennen aan de **dubbele** aanhalingstekens, die de konstante begrenzen. Deze aanhalingstekens maken zelf geen deel uit van de string. De regels waarmee de geldigheid van konstanten kan worden bepaald zijn erg ingewikkeld. In het algemeen zijn numerieken legaal als ze bestaan uit karakters van **0** tot **9**, **+**, **-** , **E** en als laate een punt '**.**' Getallen als **7a+3b** worden niet gezien als konstanten, terwijl ook schrijfwijzes als **1.2.3**, **4E2E1** (4 * 10^20), hoewel passend in de regel niet als geldige numerieken worden beschouwd. Zoals in één oogopslag te zien is, zal de enige echte boosdoener de letter 'E' kunnen zijn.
Voor strings ligt het allemaal wat gemakkelijker, want alles wat tussen de aanhalingstekens staat hoort bij de string. Alleen de niet te printen karakters, zoals Top-of-Form, Carriage-Return en ASCII-NULL zijn moeilijk te be-

kijken en kunnen wel degelijk deel uitmaken van een string.

Variabelen

We hebben al veel vaker over Basicvariabelen gepraat, met name over alle regels waaraan ze moeten voldoen om door een Basic-programma op de juiste manier te worden geïnterpreteerd. Vandaar dat we in dit geval slechts kort aandacht aan dit programma onderdeel zullen besteden.
Een Basic-variabele bestaat uit een combinatie van minimaal één letter met eventueel een achtervoegsel om de aard van de variabele aan te geven. De maximale lengte van een variabele hangt af van de gebruikte interpreter. Bij de Commodore-Basic is het zo, dat een naam langer kan zijn, dan het Basic-programme herkent. Commodore beperkt zich tot de eerste twee karakters. Normaal wordt door de interpreter echter de volledige naam aan de waarde van een variabele toegekend.
Een variabele **moet** altijd een waarde hebben in een programma. Een waardeloze variabele kan eenvoudig niet bestaan in welk computerprogramma dan ook. Standaard wordt bij de definitie van de variabelen (bijna) automatisch de waarde 0 (of "" NULL voor strings) aan de variabele toegekend. In tegenstelling tot de konstanten kan de waarde van een variabele gedurende de run van het programma echter worden gewijzigd.
Geldige namen voor variabelen zijn bijvoorbeeld:
A, I, J, X1, XX4, X%, AA$, RENTE, FRMULE%, NAAM$, N% en reeksen andere mogelijkheden. Hoe langer de naam, hoe beter herkenbaar de variabeler en even zoveel extra typewerk om het programma te schrijven. De leesbaarheid wordt echter evenredig duidelijker naarmate de lengte van de variabele-naam toeneemt. Bij bepaalde soorten dialecten van Basic, waaronder Commodore, is het illegaal om namen van Basic-opdrachten per abuis als onderdeel van een variabele naam te laten fungeren. Ik heb eens een kort voorbeeldprogramma voor Commodore-Info geschreven, waarbij voor het gemak de variabele GETAL werd genoemd. Het programma werd getest op een IBM PC, waar met GWBASIC of BASICA dit soort naamgeving volledig wordt geaccepteerd. Nog nooit heb ik echter zoveel brieven met terechte kritiek gekregen van Commodore-bezitters, die allemaal vastliepen op deze naam, waarin de opdracht GET abusievelijk was verwerkt. Gelukkig is dit type fout snel en gemakkelijk uit een Basic-listing te verwijderen, als het nodig mocht zijn, maar lastig is het wel. Want namen als **FOR**MULE, **GET**AL, E**VAL**UTATIE, A**THEN**E$, **DEF**AULT, KAL**END**ER, **NOT**ITIE, COMM**AND**O en **INT**EREST zijn bij Commodore-Basic niet te gebruiken, hoewel ze vaak exact weergeven waar een bepaalde variabele voor wordt bedoeld.
Ook de combinatie van namen die onder andere bestaan uit de letters ST of TI zullen bij Commodore voor nogal wat problemen zorgen.
We zullen proberen deze handicap zoveel mogelijk te vermijden.



Regelnummers

Naast commando's, konstanten en variabelen bestaat een Basicprogramma nog uit (vaak al facultatieve) regelnummers en operatoren.
De lijn- of regelnummers bestaan in alle interpretatieve Basics nog als een onvermijdelijk onderdeel van het programma. Er moet tijdens het runnen nu eenmaal een bepaalde manier zijn waardoor de 'vertaler' weet waarheen wordt verwezen bij een GOTO, IF..THEN of GOSUB opdracht. Wordt een listing vooraf gecompileerd, dan kunnen de regelnummers worden gemist en waar nodig door programmalabels worden vervangen. De modernere Basic-dialecten kennen deze manier van programma-opbouw. Vooral na de popularisering van de PC-familie hebben we een stortvloed van Basic-versies en compilers gekregen, waardoor het werken met regelnummers al bijna uit de tijd is geraakt. Maar interpretatief hebben we ze nog altijd als referentie nodig, en schadelijk zijn ze in geen geval.

Operatoren

Als laatste onderdeel van een programma hebben we nog de operatoren, waarmee we de uit te voeren bewerkingen in een programma aangeven. We doelen hiermee natuurlijk op tekens als **+,-,*,/,^ <,=,>** en de operatoren **AND, NOT** en **OR**.
Als we helemaal volledig willen zijn, onderscheiden we in deze operatoren twee types : Binaire en Unaire operatoren. Nu zijn dit ook weer moeilijke begrippen, maar de betekenis van beide valt wel mee. Binaire operatoren worden gebruikt om twee konstanten of variabelen samen te bewerken en één nieuwe variabele aande uitkomst toe te kennen.
'Unaire' operatoren hebben betrekking op één variabele (of konstante). Alleen de Boolse operator NOT en de tekens + en - zijn unair te gebruiken. Het unaire gebruik van operatoren zullen we in huis-tuin en keuken Basic niet al te vaak bewust tegenkomen, hoewel ook het minteken voor een getal in feite een unaire operator is. Voor de volledigheid zijn ze toch op deze plaats vermeld.
Alle operatoren kunnen worden gebruikt bij numerieke variabelen; voor strings kennen we echter maar één operator, het plusteken (+) waarmee de strings als het ware aan elkaar kunnen worden 'geplakt'. Verdere rekenkundige bewerkingen kunnen we met alfanumerieke data niet uitvoeren.

Prioriteit

Voor veel programmeurs is de volgorde waarin de verschillende operatoren wordt afgewerkt een moeilijk punt. We kunnen namelijk verschillende uitkomsten verwachten van de volgende som:

45 + 30 * 18

Als we eerst 30 bij 45 optellen wordt de som 75, dit 18 keer, dan krijgen we als oplossing 18*75 = 1350.
Doen we het anders, door eerst 30 met 18 te vermenigvuldigen, en er als laatste 45 bij op te tellen, dan wordt de uitkomst 585, een heel verschil met de vorige bewerking. De vraag wordt nu natuurlijk, welke van de twee de juiste is, want op het eerste gezicht is op geen van beide oplossingen iets aan te merken.
Nu kan het natuurlijk niet, dat uit één som, met dezelfde konstanten en operatoren twee oplossingen kunnen komen. Daarom heeft men in de wiskunde een soort voorrangsregel opgesteld, waardoor de prioriteit van de verschillende operatoren wordt bepaald. Daardoor weten we (en ook de computer gelukkig) welke bewerking voorrang heeft boven andere bewer-

kingen, zodat we in alle gevallen slechts over één correcte uitkomst kunnen beschikken.
Het is erg belangrijk deze prioriteitentabel goed in je hoofd te hebben, dat spaart een hele serie haakjes in je programma, zoals we straks zullen zien:
De voorrangsregels voor Basic luiden als volgt:

^ (exponent)
+,- (unair plun en min))
***,** / (vermenigvuldigen en delen)
+,- (binair optellen/aftrekken)
<,>,=(vergelijking, kleiner dan, gelijk & groter dan)
NOTBoolse NOT - unaire operator
ANDBoolse AND - binaire operator
ORBoolse OR - binaire operator

Eigenlijk allemaal wel recht voor z'n raap, de expressie:

4^2*4 wordt dus niet **4^8** maar **16*4**
Hoe hoger in de tabel, des te eerder wordt de bewerking uitgevoerd op de twee belendende waarden. In de voorgaande som (45+30*18) wordt dus eerst de vermenigvuldiging uitgevoerd en vervolgens de optelling gemaakt, waardoor de juiste uitkomst 585 wordt.
Let nog even op de unaire operatoren, want door deze toe te passen, kunnen we sommen als **4^-2*3**, **7-+3**, **-3342 * -2232** zonder problemen oplossen. De unaire operatoren + en - worden eerst verwerkt, waarna we een vrij simpele berekening overhouden, die alleen op het eerste gezicht wat ingewikkeld overkomt.

TRUE en FALSE

Naast de logische, of Boolse operatoren, AND, NOT en OR gebruiken we in computertalen maar al te vaak de conditie WAAR of ONWAAR, ofwel in beter computerjargon TRUE en FALSE. De waarden die aan TRUE en FALSE worden teogekend zijn in feite niet erg van belang, maar wel het feit dat TRUE altijd ongelijk is aan FALSE. De volgende programmaregels laten een TRUE/FALSE voorbeeld zien:

```
10  A = 10
15  B = 20
20  IF (A < B) THEN PRINT "A KLEINER
    B IS TRUE"
30  IF (A =>)B) THEN PRINT "A
    KLEINER B IS FALSE"
40  PRINT A < B
50  PRINT A > B
```

Uit de definitie van A en B blijkt, dat A inderdaad kleiner is dan B, waardoor de conditietest van regel 20 : **IF (A < B)** tot TRUE (WAAR) evalueert, en regel 30's IF is daardoor automatisch FALSE. De print-opdracht van regel 20 wordt dus op het scherm gezet.
In regel 40 en 50 zien we de waarde die onze computer aan TRUE en FALSE toekent.
Voor FALSE heeft iedere computerfabrikant gelukkig de waarde NUL (0) in huis. Regel 50 laat dit ook duidelijk genoeg zien. De waarde van TRUE is een andere zaak. Basic in het algemeen, IBM-Basic en Commodore in 't bijzonder wijken hierin dan ook van nagenoeg alle andere computertalen af, door aan TRUE de waarde -1 toe te kennen in plaats van de standaardwaarde TRUE=1. Vooral bij de wat 'professioneler' talen als 'C', COBOL en FORTRAN vinden standaard



TRUE=1.
De reden waarom Commodore heeft gemeend dit anders te moeten is niet geheel duidelijk, want er zijn nogal wat programma-voorbeelden te bedenken, waarbij juist het teken van TRUE meespeelt in de condities en de gevolgen. Denk bijvoorbeeld maar eens aan de datum-berekening waar we in één van de voorgaande lessen uitvoerig bij hebben stilgestaan.
De vergelijkingen van Commodore wijken wat betreft numerieken weinig af van de standaard. Bij strings gaat het echter allemaal wat de verkeerde kant op.
De vergelijking tussen "1000" en "10000" gaat nog goed, want PRINT "1000" < "10000" evalueert tot TRUE ofwel -1. Bij andere strings gaat het echter verkeerd, omdat de strings slechts vergeleken worden tot op de lengte van de kortste string. De test op PRINT "9" > "799" geeft TRUE als uitkomst, terwijl dit natuurlijk in het geheel niet juist is. Maar goed, daar moeten we maar mee leven.
Wel zijn het juist dit soort onvolkomenheden, die Basic jammer genoeg blijvend degraderen tot een taal voor uitsluitend amateurs.

Grenzen

De reden van deze fouten en onjuistheden in de verwerking ligt in de wijze waarop (Commodores') Basic met een dergelijke vergelijking tussen twee waarden omgaat. De logische operatoren (< en > in dit geval) gebruiken een 16-bit woordsysteem, dus twee bytes.
Daarom wordt TRUE, (-1) in het geheugen gehouden als $FFFF (65535), dus in feite de hoogste waarde die een dubbel-byte kan aannemen. Voor logische (Boolse) expressies wordt door dit gegeven de maximale waarden ook beperkt tussen -32768 en +32767. We kennen deze limieten waarschijnlijk al van de integer-variabelen. Zodra we buiten deze waarden komen, ook met tussentellingen bij het uitwerken van expressies, zullen we onmiddellijk een 'overflow' error kunnen verwachten. Een voorbeeld hiervan is de Basic opdracht PRINT NOT 32768. Omdat door NOT de 16-bits allemaal worden 'geflipt' van 1 naar 0 of juist omgekeerd, krijgen we bij NOT 10 bijvoorbeeld -11 ofwel (NOT X)+X = -1 gaat altijd op. Overigens is het aardig om dit eenmaal te lezen en wellicht ooit, bij het zoeken naar een vervelende, weggedoken 'bug' in je programma ineens te herinneren, dit soort werk zal over het algemeen weinig in de standaard Basic-programmatuur worden teruggevonden. Hoewel zeker in Basic de verwerkingstijd een belnagrijk gegeven vormt tijdens het runnen, zijn zeker voor de hiervoor geschetste problematiek goede, omzeilende oplossingen te bedenken, zodat we de Basic-taal om dit soort onvolkomenheden niet van de tafel behoeven te vegen.
Wel nog even terug naar de voorrangsregels bij operatoren. Deze zijn echt erg belangrijk, leer ze uit je hoofd, voorzover je het niet al wist. Let daarbij vooral op de onderste regionen. Die zorgen vaak voor de meeste, niet voor de hand liggende fouten in de software.

Functies

Over functies hebben we al eerder gesproken. Alle opdrachten in Basic, waarbij naast het keyword ook tussen haakjes een expressie moet worden meegenomen, noemen we functies. Het betekent dus een opdracht die geheel gewijd is aan de uitdrukking tussen de haken. MID$("CBAM-

SAND",5,3) wordt "SAN", en SIN(4^2) geeft
-.2879034 als oplossing. De gehele functie als zodanig evalueert tot een waarde. Een uitzondering op deze functieregel is de opdracht PEEK. Dit is geen functie in de reinste zin van het woord, maar voor het gemak is deze opdracht gegoten in de jas van functie en wordt daarom ook maar als zodanig beschouwd.

Haken

Om een expressie op een niet standaardmanier door de computer te laten uitvoeren kunnen we gebruik maken van de ronde haken '(' en ')'. Door delen van een berekening met konstanten en variabelen tussen haken te plaatsen, kunnen we er zeker van zijn, dat deze gedeeltes voor alle andere bewerkingen door de computer zullen worden aangepakt en bewerkt. Eerder hebben zagen we een expressie met twee mogelijke uitkomsten, doordat we niet op de hoogte waren van de prioriteiten. at we niet op de hoogte waren van de prioriteiten.
De uitkomst van 45 + 30 * 18. Willen we hierin eerst 45 en 30 sommeren, dan kan dit door de computer gebeuren, door dit deel tussen haken te plaatsen waardoor we het volgende bereiken: **(45 + 30) * 18** . Eerst zal nu de optelling worden gemaakt, (75)*18, waarna pas de vermenigvuldiging wordt uitgevoerd, in tegenstelling tot de standaard prioriteits-regels. In feite kun je zeggen, dat de haken bij Basic de hoogste prioriteit hebben.

Expressie

Het woord 'expressie' is vooral in deze les een aantal keren gevallen, en iedereen zal ook wel grofweg weten wat er precies mee wordt bedoeld. Het betekent een aantal variabelen, operatoren en eventueel functies bij elkaar, die samen als eenheid door de computer moeten worden behandeld in verband met een specifieke Basic-opdracht.
Om het moeilijk te maken, onderscheiden we maar liefst drie types expressies:

- **String expressies**
- **Rekenkundige expressies**
- **Logische expressies**

De **string-expressie** is een verzameling van één of meer stringvariabelen, string-functies en string-konstanten, eventueel met haken verbonden en ingedeeld, samen met de enig mogelijke string-operand (+).

Voorbeeld van zo'n string-expressie is
:
STR$(135)+"ARDNAS"+MID$(Q$+" YES",3,4)+CHR$(X)
Met een beetje goede wil is het allemaal nog ingewikkelder te maken, maar de bedoeling zal duidelijk zijn, ook al stelt dit voorbeeld niet veel voor.
MAAND$ = MID$("JANFEBMAA APRMEIJUNJULAUGSEPOKT NOVDEC",X,3)
heeft echter al heel wat meer nut als string-expressie.

Een **rekenkundige expressie** bestaat uit een verzameling van numerieke konstanten, dito variabelen en functies, samengevoegd met hulp van alle operatoren en haken, waarbij gebruik kan worden gemaakt van integer en reële variabelen. Een rekenkundige expressie moet altijd worden gebruikt in samenhang met I/O opdrachten of als toekenning aan een bepaalde variabele.
SQR(VAL(X$(2,4,7))+ KL% + ABS (SGN(X-Z)*-334)*(A>QAND17*X)
is een prachtig voorbeeld van een dergelijke expressie.



De **logische expressie** is meestal wat minder ingewikkeld, maar niet minder moeilijk te interpreteren.
Wat je hiervan op z'n minst moet weten, is dat een logische expressie maar twee uitkomsten kent, TRUE en FALSE, zoals we eerder in dit artikel voldoende hebben bekeken. Een dergelijke expressie bestaat uit een verzameling van variabelen en/of konstanten, waarbij uitsluiten gebruik wordt gemaakt van haakjes en -hoe kan het ook anders- de logische of boolse en conditionele operatoren. Dus OR, AND, NOT, <,=,> en verder niet. Een voorbeeld is :

(A < 20) OR ((B = 10) AND NOT (22=X))

Hoewel in theorie nogal verschillend van de voorgaande expressie, kunnen in Basic beide expressies gemakkelijk doorelkaar worden gebruikt, want de computer gooit de hele expressie, logisch of numeriek, of zelfs een combinatie van beide op één hoop en verwerkt de uitdrukking in dezelfde Basic-routine. Een deel van de oorzaak van de foute TRUE-waarde overigens.
Daardoor kunnen we in Basic eigenlijk heerlijk aan het rotzooien met rekenkunde en eigen fantasie, die daar volledig tegenin druist, tot groot ongenoegen van de echte logicus. Uitdrukkingen als

ON 2 +((P=Q) AND X <4) GOSUB 1000,2000

kunnen nu eenmaal niet in de wereld van de professional, maar onze Basic computer doet precies wat we voor ogen hebben met deze opdracht. Dat het daarom ook allemaal goed te praten is, geloof ik niet, maar ik moet toegeven, dat een regel als bovenstaande er verdraaid ingewikkeld uitziet.

Statement

In Basic spreken we van een statement, als combinatie van keywoord of opdracht samen met één of meer functies en expressies. Het is qua zinsbouw de kleinste korrekte eenheid waar we van kunnen spreken in een Basic-programma. **PRINT 5** is bijvoorbeeld een statement. Een statement wordt begrensd door een dubbele punt of een EOL, de afkorting voor END-OF-LINE (einde van een regel). Alle statments beginnen met een Basic-opdracht, of in speciale gevallen met een variabele. Dat is het gevolg van de opdracht LET, die in latere Basic-dialecten kon worden weggelaten. Het statement **A=34** zou dan ook feitelijk als **LET A = 34** geschreven moeten worden, en beantwoord daarmee weer helemaal aan de statement definitie.
We kennen nogal wat types STATEMENTS, waar we niet al te diep op in zullen gaan, want we hebben al genoeg theorie gehad. Wel zullen we ze stuk voor stuk kort noemen.
Als eerste de **toekennings-statements**. Deze geven aan een variabele een bepaalde (begin)waarde. LET C = 45 of LET A = TI zijn bijvoorbeeld toekennings-statements.
De operator '=' wordt in deze opdrachten heel anders gebruikt als we gewend zijn van de standaard '=' als logische en/of conditionele '='. De meeste computertalen gebruiken dan ook twee verschillende operators hiervoor, dooe bijvoorbeeld zaken als ':=',

'==' of '....' in het leven te roepen. Basic laat dit jammer genoeg achterwege, en laat de programmeur in feite klungelen met een sleutel voor twee doelen. Nu valt ook hier wel mee te leven al is het niet altijd gemakkelijk en zeker voor degenen die later nog eens met een andere taal verder willen studeren wordt het tweeslachtige gebruik van de '=' operator nog een ware kwelling.

Een tweede type statement is het **conditionele**. Alle opdrachten met IF..THEN vallen hieronder.

IF A = 10 THEN PRINT "OK" en **IF NOT X THEN END** zijn voorbeelden van conditionele statments.

De derde groep statments kennen we als **controle-opdrachten**. Deze groep wordt benut om de programma-flow in andere banen te leiden. Commando's als GOTO, GOSUB, END, RETURN en STOP horen thuis in deze categorie.

Vervolgens twee erg nauw verwante statement-types, INPUT en OUTPUT statements. deze worden gebruikt om de communicatie naar buiten te verzorgen. Zowel voor het inkomende (INPUT) als uitgaande verkeer (OUTPUT)

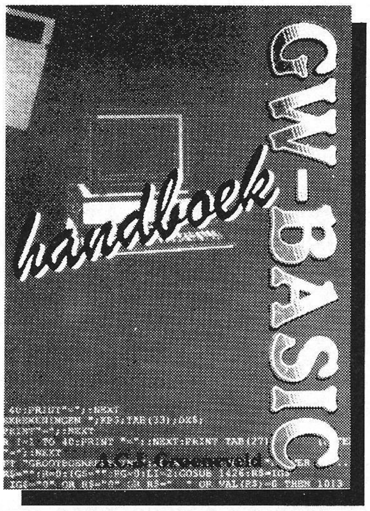
Opdrachten als INPUT, INPUT#, PRINT#, GET, READ, PRINT en dergelijke horen thuis in één van deze beide statement-groepen.

Een heel belangrijke groep statements zijn de zo genoemde '**compound**' of **blok-statements**. Deze groep zorgt, dat een hele serie statements in een blok kan worden uitgevoerd. Uiteraard kan FOR..TO..NEXT als voorbeeld hiervoor dienen. Latere (en betere) Basic dialecten hebben de mogelijkheden ten opzichte van de oude standaard drastisch uitgebreid, door opdrachten als DO..WHILE en WHILE..WEND en noem maar op. Bij Commodore vinden we helaas alleen een wat magere FOR..NEXT voor onze blok-opdrachten. De 'loop'statements vormen het eerste begin van structureel programmeren.

Dan kennen we nog de '**conversie-statements**', waarmee we het karakter van een variabele (de waarde althans) kunnen veranderen van string naar numeriek of omgekeerd. Hiervoor beschikt Basic over een groot aantal functies, zoals VAL, ASC, CHR$, STR$, INT.

De laatste en niet belangrijke groep statements wordt gevormd door de **commentaar-statements**. We zullen allemaal weten, dat alle informatie na een REM-opdracht niet door Basic zal worden uitgevoerd. Hoewel dit juist is, is het misschien leuk, te laten zien, dat het ook mogelijk is, om zonder rem toch commentaar op een programmaregel te krijgen. Zelfs hele blokken kunnen op die wijze met commentaar worden gevuld:

```
0   GOTO 100 :Begin programma op
    regel 100
10  Dit programma wordt gebruikt om
    een aardig
20  bugje in Basic te illustreren,
30  omdat ook zonder REM de regels
    die hier
40  staan probleemloos mee runnen
    met
50  het Basic-programma.
60  Ze horen daarom ook tot de
    commentaar-statements
70  zonder gebruik van de opdracht
    REM.
100 REM begin programma.
```



Tot zover de statements, en als ik het goed heb is er weinig meer van de standaard Basic regels te vertellen dan we in deze les vrij uitvoerig hebben behandeld.

Alleen over programma's en de programma-regels in hun geheel moeten we het nog even hebben.

Programma regels

De regels van een Basic programma zijn al even aan de orde geweest, maar per regel bestaat een listing uit minimaal 0 Basic-statements. Iedere regel in een programma wordt in het geheugen vooraf gegaan door een NULL-byte, zodat de interpreter het einde/begin van Basic-regels kan herkennen. Daarnaast zijn meestal een tweetal bytes gereserveerd voor het regelnummer, dat daardoor een waarde kan hebben van 0 tot 65535 terwijl ook per regel twee bytes worden gebruikt voor de schakel naar het begin van de volgende programma-regel in het geheugen. Er bestaan wat computers (zeker de oudere) die van deze regelopbouw afwijken, maar in principe kunnen we per regel de hierbij gegeven opbouw zien. Voor de duidelijkheid, we praten uitsluiten over de Basic-interpreter, en niet over gecompileerde Basic-programma's.

De programmaregel zelf kan bestaan uit strings, gecodeerde Basicopdrachten (TOKENS) , aanhalingstekens met of zonder string-konstanten, haakjes, variabelen leestekens, commentaar en DATA-statements.

Kortom, een programma-regel is een willekeurige verzameling van 0 of meer Basic-statements, gescheiden door een dubbele punt, waarbij het einde van de regel ook als scheiding tussen statements functioneerd.

Programma

Een Basic programma heeft na al het voorgaande wel een heel gemakkelijke definitie, en is niet anders dan:

Een verzameling van 0 of meer programmaregels, die alle voorzien zijn van een regelnummer en overige standaard-structuur, waarbij de opeenvolging van regelnummers het te verwachten verloop van het programma aangeeft, met uizondering van de flow die door controle-statements kan worden gewijzigd. Het programma-eind wordt intern aangegeven door drie NULL-bytes.

Meer valt niet te zeggen van een Basic-programma, al is dit juist waar het om draait.

We hebben nu wel alles gezien wat met een Basic-programma en de programmering van Basic te maken heeft. Ik ben me bewust, dat vooral deze les nogal saai en droog is geweest, maar ik hoop dat tot hiertoe het allemaal te volgen was. Dat moet bijna wel, anders zou je dit nu niet lezen.

De volgende keer beginnen we met Basic van A tot Z met de beschrijving van de eerste echte functies en commando's. Heb je intussen vragen en opmerkingen, dan kun je die schriftelijk indienen bij de redactie van dit blad, met vermelding 'Basic-probleem'. We zullen proberen voor alle vragen een oplossing te vinden, hoewel het gebrek aan tijd ons meestal voorrang laat verlenen aan problemen die voor meerdere lezers van belang kunnen zijn.

Intussen sterkte en hopelijk tot de volgende Basic-sessie.

Jan Bodzinga

Vanaf dit nummer gaat Commodore Info meer pagina's besteden aan de Amiga, zonder overigens de gebruikers van de andere Commodore computers te vergeten. Met name in de nieuwe rubriek Amiga Hoek, die door Johan en Johan wordt verzorgd, zal de lezer de hard- en software van de Amiga door en door kunnen leren kennen.

Amiga introductie

Om te beginnen gaan Johan & Johan in een aantal basisbegrippen rond de Amiga behandelen. Met hardware- software en boekbesprekingen zal er, of u nu een Amiga 500, 1000 of 2000 heeft, voor elk wat wils te vinden zijn.

Boekbespreking

De allereerste keer zullen wij direct een stapel boeken gaan bespreken van DATABECKER. Databecker is sinds jaar en dag bezig met het vertalen en uitgeven van Commodore boeken. Wie kent de INTERN nu nog niet? Dat is gewoon een begrip geworden in de Commodore wereld. En geloof ons, dat u zonder deze boeken bijna niets zult kunnen ondernemen. Het is immers een computer met geheel eigen instructies. En laten we nu niet onder stoelen of banken steken dat de bijgeleverde lektuur minimaal is! Dat is bij iedere Amiga gebruiker bekend. Om wat met uw Amiga te kunnen ondernemen zonder alle boeken over de Amiga aan te schaffen moet u dus op deze rubriek blijven letten.

Software

Dat er voldoende sofware voor de Amiga te verkrijgen is mag ondertussen ook wel bekend zijn. Veelal zijn dit spelletjes en dure comerciële programma's. Ook de allerbeste software voor de Amiga zullen wij gaan bespreken. In de volgende INFO zult u een recentie van twee C-compilers tegen komen. Zelf hebben wij net een primeur gezien op de Amiga!! Het programma wordt geleverd met hardware en is een komplete besturing voor een taxicentrale. De firma die de soft- en hardware op de markt gaat brengen is SCRAA TELECOM uit Wommels. Voor informatie kunt u het onderstaande nummer draaien/intikken: 05159-2380.

Virussen

Wanneer u nu de software wereld ook weer voor de Amiga wilt gaan verpesten door allerlei software illegaal -dus tegen de wet en strafbaar!- te gaan kopiëren zullen de pakketten steeds duurder worden. Maar er was iemand, wij noemen geen namen, die hier wat op heeft gevonden. Een VIRUS is de machtige tegenhanger van backuppend Nederland. Door nu met illegale programmatuur te werken loopt u het risico ook zo'n monster op uw diskette's te krijgen. En wanneer u een virus op een diskette heeft staan wordt het lachen. Ja, die virussen zorgen voor dolle pret dat is zeker. Amiga gebruikers kennen vast het BYTE BANDIT of het SCA virus wel. En terwijl wij dit artikel schrijven zijn er misschien al weer een paar 'schurken' die driftig een eigen virus aan het ontwerpen zijn. Dat zullen we nooit zeker weten maar het is maar een idee.

Zelf programmeren

Nee geen virus gaan zitten maken, ben je gek geworden zeg! Ook niet al was het maar om uw naam door de hele wereld te sturen. Er is een betere manier om wat met uw kennis te doen. Maak eens een leuk programma en stuur dat ook eens naar ons op. Wanneer het van goede kwaliteit is, krijgt ook heel Nederland het te zien. In de nieuwe Amiga hoek zullen wij in een volgend nummer van Commodore INFO ook van start gaan met een C-cursus voor de Amiga. Natuurlijk zullen wij simpel starten en later op volle toeren door de Amiga heenvliegen. De Amiga-hardware zullen wij stap voor stap doornemen. Net als in de TRIPLE 128 rubriek zal er voor de Amiga een artikel komen voor de diverse hardware onderdelen. Zo zullen er besprekingen komen over de externe diskdrive, aansluiten van de 1901 monitor aan de Amiga. Wist u wel dat de 1901 monitor ook op de Amiga aangesloten kan worden? Maar dat is nog toekomstmuziek en nog even geduld hebben dus.

Vragen en opmerkingen

Voor vragen en opmerkingen kunt u vanaf nu bij de Amiga corner terecht. Maar dat moeten wij wel heel voorzichtig zeggen want anders verzuipen wij straks in de postzakken. Omschrijf uw probleem en/of opmerking a.u.b. wel heel goed. Voordat uw brief door ons zal worden beantwoord moet u wel even geduld hebben. Lees ook het stuk op pagina 21 in nummer 3 eens door. Hierin kunt u lezen op welke manier u uw brief kunt insturen. Wanneer er leuke brieven bijzitten zullen wij die in een volgend nummer van Commodore INFO publiceren.

De gebruikte lektuur

De door ons gebruikte lektuur bestaat hoofdzakelijk uit de DATABECKER reeks. Verder nog een stapel uit de ADDISON-WESLEY reeks. Onder diverse titels kunnen wij dan zo nu en dan even een boekje raadplegen.

De gebruikte hardware

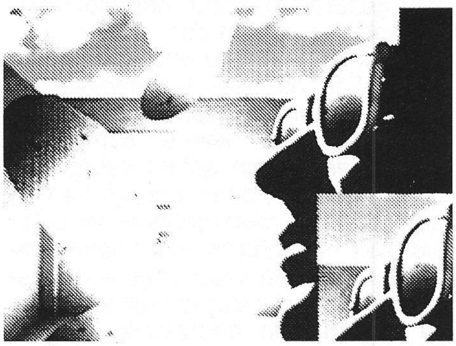
De door ons gebruikte hardware bestaat uit een Amiga 500 een Amiga 1000 en zelfs een Amiga 2000. U ziet het kan dus niet op. Wij zijn druk bezig met communicatie lijnen aan te leggen tussen de verschillende computers. In de TRIPLE 128 rubriek kunt u dit keer dan ook lezen hoe u een NULL MODEM kabel kunt maken. Met deze kabel kunt u tussen twee Amiga's en/of een andere computer gaan communiceren. Aan ons zal het dan ook niet liggen. Ook voor het geld hoeft u het niet te laten. Maar bekijk het zelf maar eens. In ieder geval hopen wij u met de introductie van de nieuwe Amiga corner een heel eind op weg te helpen. Voor ideeën kunt u ons natuurlijk ook even schrijven.

JOHAN & JOHAN.

Easyl

Nieuw op de Nederlandse computermarkt is de Easyl, een drukgevoelige professionele tekentafel die de grafische kwaliteit van de Amiga volledig kan benutten. Ideaal voor ontwerpers, tekenaars die maar niet echt kunnen wennen aan het strak tekenen met een muis of een joystick. Op de "gewone" manier met pen, potlood, lineaal maar met de computer. Tekeningen die u op dit tableau maakt verschijnen dan ook op het beeldscherm. Hiermee ligt de weg open om goed te kunnen tekenen en toch de beelden met de Amiga te kunnen bewerken. We beschikken nog te kort over de tekentafel om alle mogelijkheden te benutten. We wilden u dit nieuws niet onthouden, maar in een volgend nummer zullen we uitgebreid op deze tekenmogelijkheid terugkomen.
De Easyl wordt in Nederland op de markt gebracht door de firma Cat & Korsh International (010 - 4507696) en kost voor de Amiga 500 *f*899,-- voor de Amiga 1000 *f* 999,-- en voor de Amiga 2000 *f* 1099,--

Nieuw deel in DeLuxe-reeks



De bekende reeks van DeLuxe pakketten voor de Amiga is uitgebreid met DeLuxePhotolab. Dit pakket van Electronic Arts is niet alleen een tekenprogramma dat werkt in elke grafische modus van de Amiga, inclusief HAM. Daarnaast bevat het een beeld-verwerkend systeem en een printprogramma dat het mogelijk maakt de 'kunst' af te drukken tot een maximale grootte van 3 bij 3 meter. De grootte van de tekeningen in de computer wordt beperkt door het beschikbare geheugen, en kunnen tot 20 keer groter zijn dan het Amiga-scherm. Het pakket is compatibel met de andere programma's uit de DeLuxe serie en de IFF standaard.

AmEx Magazine

AmEx Magazine is een nieuw tijdschrift op diskette. Het gehele tijdschrift bevat niet minder dan drie diskettes vol met informatie. Al bij het opstarten van de eerste disk valt op, hoe mooi de kwaliteiten van de Amiga kunnen worden benut. Een menu geeft in het kort de informatie die er op de diskette is opgeslagen. Een keuze maken, met de muis de desbetreffende regel aanwijzen, meer is er niet voor nodig om alle aanvullende informatie te verkrijgen. Onder aan het scherm staan een aantal blokjes. Quit en help spreken voor zich. Het aanklikken van het 'read' blokje is voldoende om alles over het gekozen onderwerp te weten te komen. De andere blokjes, up en down laten u door de inhoudsopgave bladeren. Eén keuzeblokje mis ik, en dat is het blokje dat bij aanklikken de tekst niet naar het scherm stuurt maar naar de printer. Disk 1 geeft de noodzakelijke informatie, terwijl disk 2 en 3 de demo's en programma's bevatten. Het grote voordeel van dit systeem is dat er, in plaats van dat er een spel beschreven moet worden, er een demo op het scherm te zien is die meestal een veel beter beeld van de software geeft.
Nu moet dit geheel niet overdreven worden, een uitleg voor de niet ervaren gebruiker mag zeker niet worden vergeten. Bij dit programma is een duidelijke Nederlandse gebruiksaanwijzing zeker op zijn plaats om te vertellen hoe het programma moet worden op gestart. Alles bij elkaar is dit "tijdschrift" zeker de moeite waard. Het is een leuke, leerzame manier om aan de nodige utilities en kennis te komen. Een redelijke kennis van het (computer) engels is zeker gewenst.

Empire

Maker Interstel noemt de ruimtesimulatie Empire het 'Oorlogsspel van de eeuw". Het doel van het spel is de vernietiging van de vijand op een nog vrijwel onbekende wereld ergens in de ruimte, om zo de totale heerschappij over die wereld te veroveren. In het spel wordt men William P. Brown, kapitein van de U.G.A.S. Britannia. Het Krellan-empirium dringt in snel tempo het gebied van de alliantie binnen, alle hulpeloze planeten die ze op hun pad vinden vernielend. Empire is een strategische simualtie van globale oorlogsvoering en opbouwen van wereldheerschappijen tussen twee tegengestelde krachten.
Electronic Arts distribueert het programma in Europa, het is verkrijgbaar voor de Amiga, de Atari ST en de PC.



Powerdrome

Een geheel andere simulatie, maar wel in de ruimte spelend, is Powerdrome, waar je een racepiloot bent die tegen vliegers van vier andere melkwegstelsels moet uitkomen in race om de Cyberneufe trofee. Met 3D grafics en realistische motorgeluiden kan men tegenstanders langs zien razen, crashen en botsen. Het is mogelijk om met een speciale dual-datalijn tegen een andere speler op een andere computer te strijden. Ook dit spel is van Electronic Arts en is er voor Atari ST en Amiga.

Casino Roulette

CDS Software brengt voor de Atari ST en de Amiga Casino Roulette, een roulettespel dat realistisch is opgebouwd. Op een Franse of Internationale tafel krijgt men de zelfde kansen als in het echte gokspel, en als het geld op is, is het spel afgelopen, net zoals in de echte wereld. Alleen wordt er nog niet met echt geld gespeeld. Er zijn in het spel wat mogelijkheden voor analyse van de afgelopen spelen, zowel wat de verrichtingen van de spelers als het rollen van de balletjes betreft.

F/A-18 Interceptor

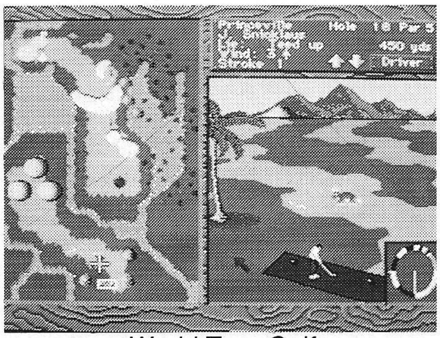
Opnieuw een vluchtsimulator, nu een F-18 of F-16 gevechtsvliegtuig dat in de buurt van de San Fransisco Bay Area vliegt. Men kan in de vliegschool

meedoen of zelf in een vrije vlucht proberen onder de Golden Gate brug door te vliegen, de Transamerica Pyramide bombarderen of de beowners van Alcatraz schrik aanjagen door laag over te vliegen. Een spel van Electronic Arts voor de Amiga

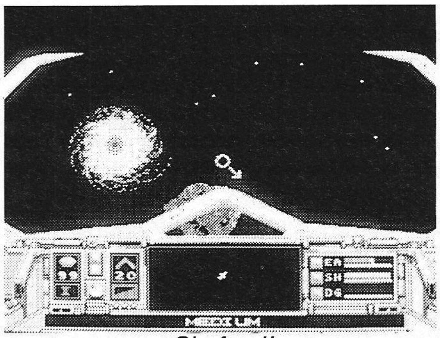
Communicatie

A-TALK III is een nieuwe versie van A-Talk PLus van Oxxi, waarin meer protocols zijn opgenomen, er is een uitgebreide Script-taal met een lesmodule om communicatie-taken eenvoudig te kunnen automatiseren. Er is een telefoonnummerlijst van 60 nummers, de snelheid van overbrenging onder Kermit is verdubbeld, het uitgebreide toetsenbord van Amiga 500 en 2000 wordt ondersteund, vreemde lettertekens kunnen worden toegepast en er worden meer dan 10 modemtypes ondersteund.

Twee zeer populaire spellen voor de Amiga zijn Skyfox II en World Tour Golf. Zelfs de redakteuren van Commodore Info maken zich nog wel eens schuldig aan een ruimtereisje, en op de vrije zondag wordt ook regelmatig een balletje tgeslagen. Van Electronic Arts kregen we een aantal foto's van beide spellen, die we hier toch willen laten zien.



World Tour Golf



Skyfox II

Krystal - Superspel?

In een artikel in Amiga User International wordt melding gemaakt van het potentiële superspel van het jaar, Krystal. Dit spel wordt in augustus door Prism Leisure Corporation op de markt gebracht, en er is sprake van een unieke aanpak op het gebied van arcade, strategische simulatie en avontuur. We zijn benieuwd.

GIGANOID

De firma Microdeal blijft zeer regelmatig nieuwe spel-programma's produceren. Dit keer een spel dat niet uit de eigen keuken komt maar gemaakt is door Swiss Computer Arts, waarvoor Microdeal de distributie verzorgd.

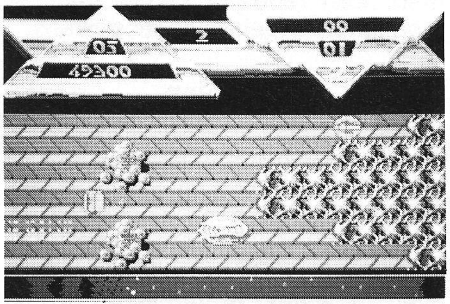
Giganoid is een super breakout. Eenvoudig gezegd: met een balletje een steeds weer ingewikkelder gebouwde muur van kleurige steentjes, door het raken van de stenen met het balletje, afbreken. Onder in beeld moet de speler met een naar links en rechts beweegbaar 'plankje' het balletje tegenhouden zodat het weer richting muur stuitert om het breekwerk te vervolgen. Is een muur volledig afgebroken dan gaat de speler door naar het volgende, moeilijker, niveau.

Dit soort spellen is voor veel spelpookvirtuozen zonder meer verslavend. Vooral als de ontwerpers er in slagen steeds weer andere en verrassende niveau's in het spel te plaatsen. Zo kan in dit programma, na het succesvol afbreken van zes muren, een nieuw niveau worden gekozen. Vier demonen staan klaar om een balletje in je richting af te vuren.

Zoals te verwachten was, demon nummer vier schiet de bal met een verwoestende snelheid door het veld. Vergeet die dus maar even en richt je als beginner liever tot nummer een of twee. Na nog een handvol, steeds weer ingewikkelder, muurtjes wordt de terugschietende tegenstander ook steeds moeilijker te verschalken.

Giganoid is de waardige opvolger van Arkanoid en als zodanig natuurlijk weer moeilijker en verrassender. De 3.5 inch-disk voor Amiga-computers kost 60 gulden.

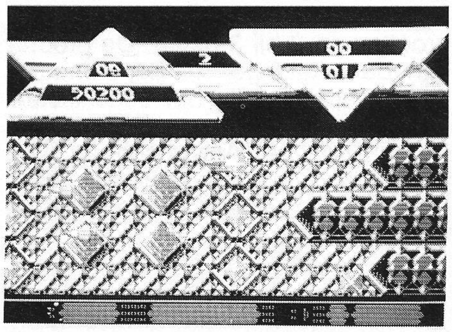
RETURN TO GENESIS



Firebird, een van de Engelse softwarehuizen onder de paraplu van British Telecom, brengt het spel 'Return to Genesis' uit. Een zeer snel ruimte schiet-spel voor Amiga computers. Zoals bij de meeste computerspelletjes zit ook bij dit programma een verhaaltje bijgesloten. Zo'n verhaaltje moet de speler in de juiste stemming brengen maar meestal slaat zo'n vertelsel nergens op. Zo ook in dit geval. Het spel zelf is voor de liefhebbers beslist de moeite en het geld waard. De bedoeling van het spel is om van verschillende planeten (mooie schermvelden) twaalf door de vijand gevangen genomen geleerden te bevrijden. Tijdens dat bevrijden wordt de speler uiteraard danig op de ruimtescheepshuid gezeten door een handvol fanatiek schietende vijanden.

Een van de mooiste onderdelen van dit spel is de zeer beweeglijke en doorzichtige ondergrond van futuristische objecten in twee en soms zelfs drie scherm-lagen achter elkaar. De lagen bewegen onafhankelijk van elkaar van links naar rechts en weer terug over het scherm. Het zogenaamde 'scrollen'. Je vliegt met je ruimtescheepje regelmatig achter en onder de nu en dan doorzichtige lagen. Soms bots je tegen verende obstakels die je ruimtescheepje terugkaatsen, helaas meestal richting vijand.

Onze spel-specialist is in vijf levels tot een score van ruim 68.000 punten gekomen. (Na een paar dagen spelen en weinig slaap). Er zijn tien levels dus de echte spel-fanaten kunnen hun lol

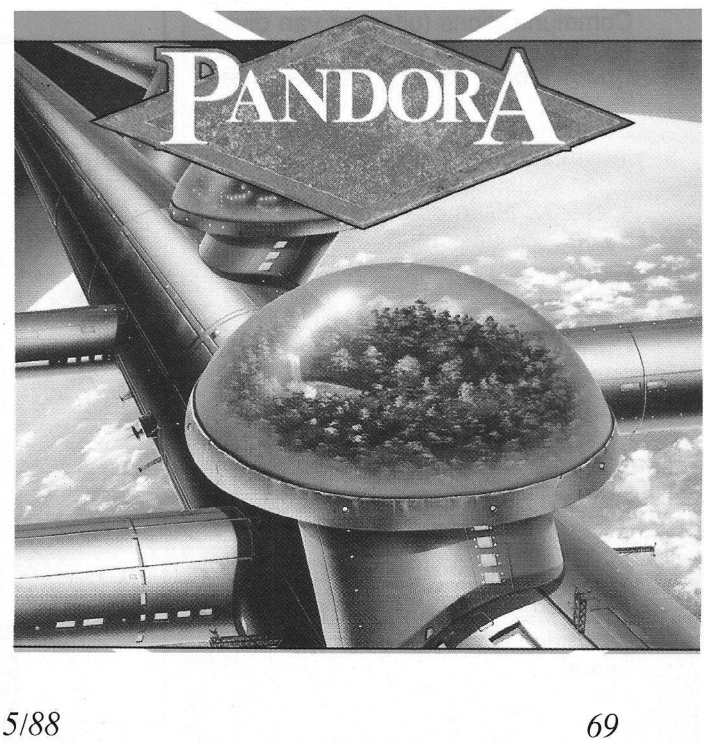




op!
Zowel grafisch als wat geluid betreft is uitstekend gebruik gemaakt van de mogelijkheden van de Amiga computer. Het geluid is echt prima, maar gaat zoals bij de meeste spellen toch na verloop van tijd vervelen. Een druk op de F2-knop van uw computer schakelt de achtergrond-muziek uit. Gelukkig blijft het geluid van de prachtige explosies, die bij het raken van de vijand een tevreden gevoel opwekken, wel te beluisteren. Met het indrukken van de F1-knop start je het spel, bijvoorbeeld na te veel gemaakte fouten, opnieuw bij het begin.
Het zijn technische geleerden die je als speler moet bevrijden. Na hun bevrijding mag je beschikken over de uitvindingen die enkele van hen hebben gedaan. Zoals het jammer genoeg bij een flink aantal geleerden in de 'echte' wereld ook het geval is betreffen de uitvindingen in dit spel betere wapens. Daardoor zal het beschieten van de vijand na sommige aan boord genomen geleerden nog soepeler gaan.
Een zeer 'snel' spel voor Amiga-computers voor 80 gulden.

PANDORA

Tweehonderd jaar geleden is het ruimteschip Pandora (tevens de titel van dit spel) van de aarde vertrokken. Volledig door een supercomputer gestuurd en met een bemanning van mensen en robots, moet het ruimtevaartuig onderzoeken of er buiten de aarde, diep in de onontdekte uithoeken van de ruimte, leven is.
Plotseling blijkt het enorme vaartuig veel eerder dan de bedoeling is met een enorme snelheid weer richting aarde te koersen. Bezorgde geleerden hebben de indruk dat er iets mis is met de boordcomputer, of misschien hebben buitenaardse wezens de macht over het schip overgenomen.
Er wordt een heel ervaren en intelligente ruimtespeurder op weg gezonden om de Pandora te onderscheppen. Wie anders dan jij kan in dit machtige ruimteschip onderzoeken of alles wel in orde is?
Eenmaal aan boord zal je vermeende intelligentie zwaar op de proef worden gesteld. Via de speelpook-besturing begin je je onderzoek. Er zijn nogal wat figuren die de speurtocht naar onregelmatigheden proberen te verhinderen. Een duidelijke aanwijzing dat er beslist iets mis is aan boord!
Het is de bedoeling zoveel mogelijk, door de Pandora in de ruimte verzamelde objecten, naar de aarde te zenden zodat ze daar kunnen worden onderzocht. Met verschillende zogenaamde ID-pasjes kan onze ruimtedetective gebruik maken van de vele boordcomputers of toegang krijgen tot beveiligde ruimtes om die te onderzoeken.
Er zal zo nu en dan een flinke knokpartij nodig zijn in het prachtige interieur van het ruimtevaartuig om onwillige types tot rede te brengen. Heb je zo'n type verslagen dan is het nuttig zijn zakken na te zoeken op bruikbare zaken. Niet bepaald ethisch maar ja, de veiligheid van moeder aarde staat op het spel zullen we maar zeggen.
Het zal snel duidelijk worden dat het niet de bedoeling is om iedereen die je pad kruist tegen de grond te timmeren. Zo nu en dan zal een vriendelijk figuur je vragen om 'iets' aan hem te geven, waarna zijn hulp je weer een stapje verder kan helpen. De zogenaamde vechtrobots kun je de eerste tijd beter ontwijken. Ze zijn in het begin niet te verslaan!
Het is bij veel spellen zinvol om na het starten van een spelschijfje de jeuk in de vingers enige tijd te laten voor wat het is. Vaak is er een demonstratie van het spel of heeft iemand moeite gedaan een aardig stukje computermuziek te componeren. Zeker in dit geval zouden wij willen aanraden de vuurknop even met rust te laten en eens rustig naar de muziek te luisteren die na de start van deze schijf uit je computer komt.
Pandora is een prima spel voor de Amiga 500 en kost 80 gulden.

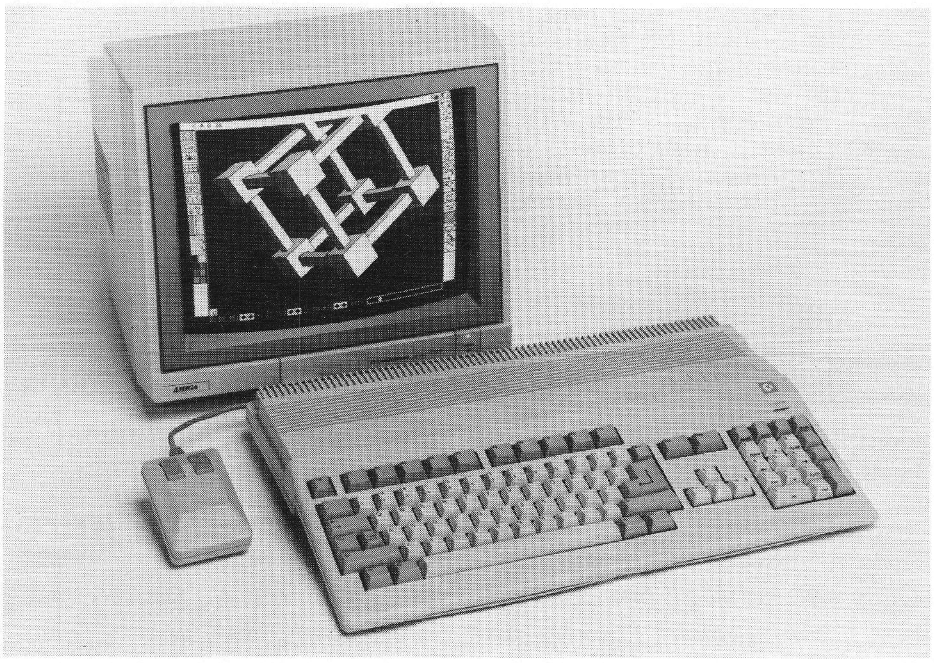


Wanneer u eenmaal een Amiga in huis heeft staan wilt u er vast mee leren werken. Waarom anders heeft u zo'n aanslag op uw beurs gedaan? Juist om er profijt van te hebben. Of het nu om tekenen of programmeren gaat uw Amiga kan het allemaal aan. Zou het dan niet de hoogste tijd worden om eens met uw aanwinst te gaan werken? Wij zijn er klaar voor, u ook? Daar gaan we dan en veel plezier ermee.

Beginnen met de Amiga

Ook voor personen die nog geen Amiga hebben, maar er wel over zitten te denken, is deze reeks van artikelen zeer aantrekkelijk. Lezen lezen en nog eens lezen. Wie weet krijgt u wel een goed idee voor een zelf gemaakt programma. En wie weet brengt dit programma nog enige financiële voordelen met zich mee. Maar goed, daar zult u zelf maar voor moeten zorgen.

Wij zijn er alleen om de informatiestroom naar de lezer te vergroten. Hoe meer u leest des te meer u zult kunnen ondernemen met uw computer. Natuurlijk geldt dit niet alleen voor de Amiga maar ook voor elke andere computer. Wij zullen in de volgende nummers wat pagina's publiceren met veel starters informatie. Deze informatie helpt de startende Amiga gebruiker de goede weg in te slaan. Stap voor stap zullen wij alles gaan doornemen. Bovendien zult u uw Amiga vocabulaire verder kunnen uitbreiden.



De eerste stap

De allereerste keer dat u de Amiga zult aanzetten, dient u wel het een en ander over de Amiga te weten. Raadzaam is dan ook eerst de handleiding eens goed door te nemen. Zo maakt u nu gebruik van 3.5 inch diskette's en geen 5,25 inch. U dient gebruik te maken van een wat betere kwaliteit diskette. Verder ook een dubbelzijdige diskette en geen enkelzijdige! Bij aanschaf van diskettes dient u hier op te letten. Omdat de enkelzijdige diskettes aan de, laten we voor het gemak maar zeggen, achterkant van de diskette niet met data kan worden beschreven. Het zou kunnen gebeuren dat de data op de, door de fabriek niet gegarandeerde kant, veloren gaat!! Weet u niet welke diskette's u moet aanschaffen vraag uw verkoper dan even om goede raad! Vooral dat laatste dient u niet te vergeten. Betaal ook niet te veel voor de diskettes, kijk dus bij meerdere winkels wanneer dat mogelijk is. Elke gulden (of franc) is er één nietwaar! In onze woonplaats zijn een paar winkeliers die elkaar het leven zuur proberen te maken door telkens weer de diskette goedkoper te maken. Lang zal het dan ook niet meer duren voordat een van beide verdwenen is. En dit zou dan de tweede worden in onze woonplaats. Aan 10 diskettes zult u in het begin wel genoeg hebben. Voor het gemak gaan wij er in deze tekst van uit dat u de beschikking heeft over minimaal een Amiga 500 met bijbehorende monitor. U kunt gebruik maken van of een TV of monochroom monitor. Bij het gebruik van een TV dient u wel een modulator aan te schaffen. Als kleurenmonitor kunt u de Amiga 1081 of 1084 gebruiken. Natuurlijk kunt u ook andere geschikte monitoren aanschaffen, maar vraag dan wel duidelijk om een geschikte monitor. Neem geen genoegen met een monitor zonder garantie, of de monitor moet navenant laag geprijsd zijn. Vraag desgewenst of de verkoper het geheel aan een Amiga wil koppelen en u ziet snel genoeg of het wil samenwerken. Op deze manier hoeft u niet nog eens terug. Vergeet ook nooit een passende kabel te vragen om de monitor aan de Amiga te kunnen aansluiten. Het zou naar zijn wanneer u een heel weekend niets kunt doen omdat de kabel niet goed is. En wanneer dan alle kabels en verdere aansluitingen op de juiste manier, volgens de handleiding, zijn aangesloten kunnen we beginnen.

Het aanschakelen

Uw Amiga aanzetten is een kwestie van de juiste stekker in de wandkontaktdoos te steken en de juiste schakelaar om te zetten. Schakel de monitor in en u ziet nu hoogst waarschijn-

lijk een fraaie afbeelding op uw scherm. Om de 2 à 3 seconden hoort u een lichte tik. De Amiga vraagt om de Amiga WORKBENCH diskette. Bij elke tik initialiseert het interne systeem de diskdrive om te 'kijken' of er al een bruikbare diskette in het gleufje is gestopt. Is dit positief dan wordt vervolgens onderzocht of het ook om een BOOTABLE disk gaat.



Bootdisk

Wanneer er nu een DOS (BOOTABLE disk) in de drive zit wordt het AmigaDOS ingeladen. En wanneer nu blijkt dat er nog geen DOS disk in de drive zit vraagt het systeem om een andere diskette (AmigaDOS staat voor Disk Operating System). Dit steeds weer opnieuw initialiseren van de diskdrive heet ook wel 'vreten' of 'happen'. Plaats nu maar eens de WORKBENCH diskette in de drive. U krijgt een copyright melding, en wat andere, hier verder onbelangrijke informatie, op uw scherm te zien. U hoort nu de diskdrive zijn werk verrichten, DOS inladen. Vergeet niet dat u NOOIT een diskette in of uit de drive stopt of haalt als de drive nog aktief is! Een bezoekje bij de computerdokter is dan niet geheel onnodig!

Workbench

Wanneer de drive klaar is met laden is de WORKBENCH geactiveerd. U ziet een pijltje op uw scherm, dat u met de muis kunt bewegen. Vanuit de WORKBENCH kunt u programm's inladen door de ICONS aan te klikken. U gaat gewoon met de pointer op de afbeelding van het betreffende programma staan en klikt twee keer heel snel achter elkaar de afbeelding aan. Door bij het opstarten van de Amiga op <CONTROL-D> te drukken en vervolgens load wb in te tikken komt u in de CLI. CLI staat voor Command Line Interface. Vanuit CLI heeft u niet die mooie plaatsjes die u wel in de WORKBENCH heeft. Vanuit de CLI moet u commando's intikken. Lees het gedeelte waar wij de CLI instructie's uitleggen maar eens door. Deze afdeling zult u voorlopig nog wel tegenkomen.

JOHAN & JOHAN.

Zoals u als Amiga gebruiker wel zult weten zijn er een heleboel aparte commando's te gebruiken. Maar welke zijn dat nu en nog belangrijker: wat doen die. In dit artikel kunt u een start vinden van de AmigaDOS CLI cursus. De schrijvers van de cursus zijn JOHAN & JOHAN

Binnenin AmigaDOS

FORMAT

Dit commando wordt gebruikt voor het formatteren van een diskette. Dit houdt in dat de diskette verdeeld wordt in 80 tracks (cilinders). Een track bestaat uit 2 sporen, dit omdat de Amiga diskdrive een dubbelzijdige diskdrive is. Deze schrijft dus aan elke kant van de diskette een spoor. Elk spoor is weer onderverdeeld in 11 sectoren. Op elke sector kunnen 512 Bytes geschreven worden. De opslagcapaciteit is nu vrij makelijk te berekenen. We hebben namelijk 80 tracks maal 2 sporen maal 11 sectoren maal 512 bytes is 901120 Bytes. Aangezien een kiloByte uit 1024 Bytes bestaat moet het totale bedrag nog eens door 1024 gedeeld worden, we komen dan uit op een totaalcapaciteit van 880 KByte per diskette. Het formatteren van een diskette gaat als volgt:

FORMAT DRIVE <disk> NAME <naam> [NOICONS].

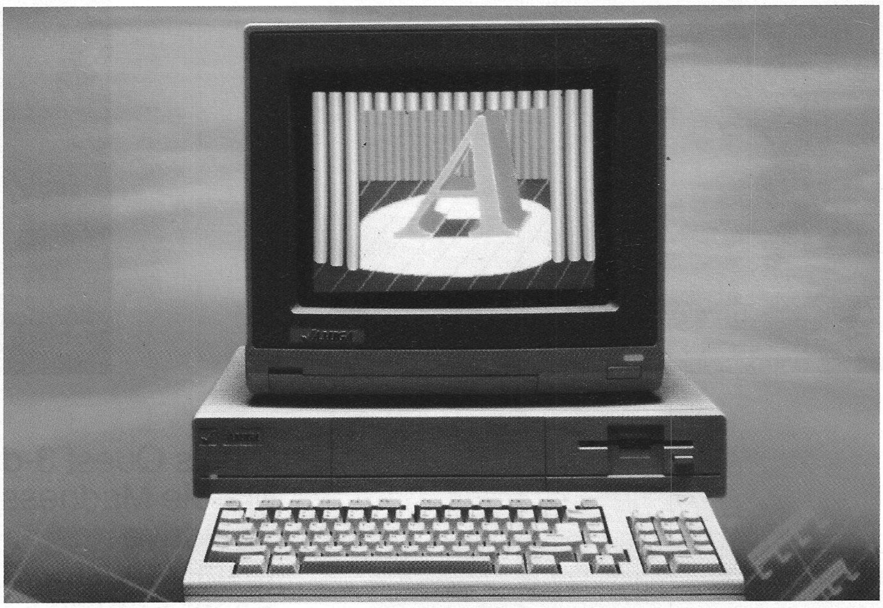
De toevoeging NOICONS is optioneel, dwz. dat het niet perse nodig is om deze erachter te zetten. De functie van het keyword is, dat de gegevens voor de Trashcan niet op de net geformatteerde schijf worden weggeschreven. Willen we nu een diskette formatteren in de standaard diskdrive (df0:) met de naam 'test', dan doet u dit als volgt:

FORMAT DRIVE df0: NAME test

Na het invoeren van deze instruktie wordt het commando FORMAT van disk gelezen. Pas als dit gebeurd is zal de volgende melding in het CLI-venster verschijnen:

'Insert disk to be Initialized in drive df0: and press Return'

Nu kunt u de te formatteren diskette in de diskdrive stoppen. Denk er wel om dat een disknaam niet langer mag zijn dan 30 leestekens. Wij raden u aan hier niet te veel gebruik van te maken, want u loopt namelijk de kans dat een lange naam in de workbench een andere naam overlapt. Een naam mag geen spaties bevatten, dit geldt trouwens voor de gehele CLI. Wel mogen er spaties gebruikt worden als het gehele onderdeel van de instructie tussen aanhalingstekens is geschreven, bijvoorbeeld: DIR "df0: basic listings". Toch is het verstandiger dit niet te vaak te gebruiken omdat het lastiger is een dergelijke naam te gebruiken. Het is veel beter gebrui k te maken van het 'underline' teken.



DIR

Met de instructie DIR kunt u een inhoudsopgave opvragen van een diskette, harddisk of ramdisk. Voor het opvragen van een directory geldt de volgende syntaxis:

DIR DIR,OPT/K

U bent bijvoorbeeld op zoek naar de file grafieken in de subdirectory basiclistings op een diskette in diskdrive df0:. Dit doet u dan als volgt: DIR df0:basiclistings/grafieken. Om de gehele directory van de schijf te bekijken gebruikt u: 'DIR opt a'. De optie 'a' staat voor ALL. Op het eerste gezicht lijkt de output een geordende chaos maar de structuur blijkt toch zeer eenvoudig te zijn. Een voorbeeldje. Stel u heeft een schijf met de directory basicdemos. Deze is dan weer onderverdeeld in grafiekdemos en geluiddemos. DIR opt a geeft dan weer:

basicdemos (dir)
geluidsdemos (dir) - hierin staan geluid demos.
grafiekdemos (dir) - hierin staan grafische demos.
.. - eventuele andere programma's

Met 'DIR opt i' bent u in staat een directory langzaam te bekijken. De 'i' staat voor INTERACTIVE. Deze optie zorgt ervoor dat de uitvoer na elke naam stopt en er een vraagteken achter komt te staan. Wanneer u een vraagteken invoert krijgt u de mogelijke opties op uw beeldscherm te zien. U kunt met een <RETURN> verder of met <DEL> het vermelde bestand wissen. Het is echter niet mogelijk een bestand te wissen dat een of meerdere subdirectories bevat. Als achter de gegeven naam de toevoeging 'DIR' staat kunt u door op 'E' (enable) te drukken deze directory activeren. U krijgt nu automatisch de inhoudsopgave van deze subdirectory. Het is mogelijk de vorige directory te vervolgen door op 'B' (kort voor back) te druk-

ken. Met het commando 'T' (dit is de afkorting van type) kunt u de ASCII-bestanden lezen. Dit heeft alleen betrekking op tekstbestanden. Gebruikt u het bij normale programma's of CLI routines dan zult u allemaal rotzooi op uw scherm zien. Dit is echter met het CLI-commando TYPE weer op te lossen. Meer hierover bij de beschrijving van dit commando. Met <CONTROL-C> kunt u de T optie van DIR op elk gewenst moment afbreken. Als er dan nog rotzooi verschijnt kun u met <CONTROL-O> de Amiga terugbrengen in zijn normale tekenset. Om de directory te verlaten kunt u Q (quit) intoetsen. Het is mogelijk om de gehele directory interactief te bekijken met het commando 'DIR opt ai'. Het is dus mogelijk de opties bij elkaar te zetten. Het DIR commando beschouwt ze als aparte opties. Als u een normale directory trekt kunt u de uitvoer van een directory stoppen door een willekeurig toets in te drukken. Pas wanneer u een <RETURN> geeft of wanneer dit teken wordt verwijderd met een <BACKSPACE> komt de uitvoer weer op gang. Verder kent DIR nog een 'redirection teken'. Het is mogelijk om de uitvoer van DIR in een file op te slaan. Probeer het volgende regeltje: DIR inhoudsopgave opt a. In de file inhoudsopgave zal nu de gehele inhoudsopgave van de schijf te zien zijn.

CD

Dit staat voor 'change directory'. Dit betekent dus dat u met dit commando van actuele directory kunt verwisselen. Alle CLI commando's zoeken, als er geen directory opgave is gedaan, in deze directory. Bijvoorbeeld, CD basicdemos DIR is hetzelfde als: DIR basicdemos. De instructie wordt op de volgende manier gebruikt: 'CD <directory>'. Met DIR is het nu mogelijk deze directory te bekijken. Met de instructie 'CD df0:' komt u weer terug in de rootdirectory van diskdrive df0: Het kan ook voorkomen dat u niet meer weet welke directory actueel is. Om dit te weten te komen typt u in CD. Op het scherm zal dan de naam van de actuele directory verschijnen.

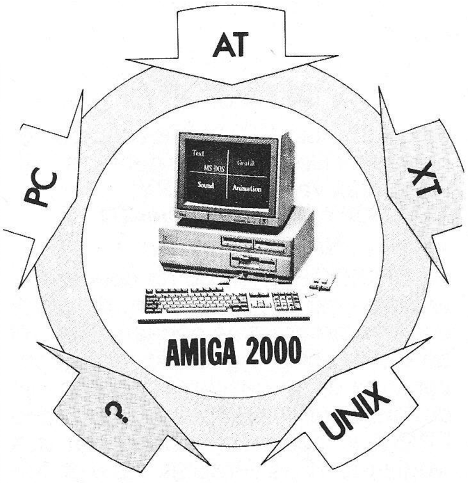
MAKEDIR

Hiermee kunt u een directory aanmaken. De directory wordt altijd een niveau lager aangemaakt dan de bestaande directory. Dit wil zeggen, de directory die u aanmaakt zal binnen de actuele directory verschijnen. Maar weer even terug naar de voorbeelddirectory. Stel u bevindt zich dus in de directory basicdemos. U maakt nu een directory aan met de naam 'test'. Deze directory zal nu binnen de directory 'basicdemos' staan. Met DIR roept u deze dan op de volgende manier aan: DIR basicdemos/test. De instructie voor het aanmaken van een directory ziet er als volgt uit: 'makedir df0:basic/info'. Er word nu in de directory 'basic' een subdirectory 'info' aangemaakt. Met copy kunt u dan de directory verder uitwerken, maar dat is van latere zorg.

DELETE

Met deze functie kunt u een bestand of een directory wissen van de vermelde diskdrive, dit geldt ook voor een ramdisk: 'delete df0:c/list'. In de subdirectory 'c' zal nu de file 'list' op diskdrive df0: worden gewist. Een subdirectory kan op de volgende manier gewist worden: 'delete df0:basic/info all'. Het is ook mogelijk om met een delete meerdere files in een keer te wissen (max. 9). Tussen elke te deleten file moet dan wel een spatie:

delete df0:c/clock
df1:s/startup-sequence



Een andere manier om meerdere bestanden of directories in een keer te wissen is door met behulp van de zogenaamde jokers (# en ?). Stel op een diskette staat een hele waslijst aan test programma's, test1 tot en met test12, deze kunnen nu in een keer gewist worden:

'delete df0:test#?'

Het '#?' zorgt er voor dat alles wat met test begint wordt verwijderd. Pas echter wel op met het gebruik van jokers, u zult niet de eerste zijn die zijn net aangemaakte bestand per ongeluk 'wist' door het gebruik van deze jokers. Door de toevoeging 'q' (quiet) voorkomt u dat de naam van de file die gewist wordt, afgebeeld wordt op het beeldscherm.

COPY

Wellicht een van de meest gebruikte commando's vanuit de CLI. Met dit commando is het mogelijk bestanden of zelfs gehele directories te copiëren. De syntax van het commando luidt;

COPY <bron> <to> <doel> opties

De bron en het doel mogen gelijk zijn, dat wil zeggen je mag op hetzelfde device, of in dezelfde directory kopieren. Ook bij COPY geldt weer het gebruik van jokers. Vooral bij programma's die files met een nummer extensie wegschrijven, bijvoorbeeld, superbase die de files met een dergelijke extensie wegschrijft, is deze mogelijkheid zeer handig. Een voorbeeld. Stel u werkt met superbase, en u wilt van uw bestanden een backup maken. U maakt een kleine batch-file aan die als volgt kan luiden:

superbase ;bijvoorbeeld de aanroep van superbase.
copy programmas.#? to df1:
echo "bestanden gekopieerd"

Na deze operatie zullen alle bestanden waarin de naam 'programmas' aan het begin voorkomt gecopieerd zijn. De opties die COPY ondersteund zijn; **ALL**: dit betekent dat alle files die in de directory, die u als brondirectory opgegeven heeft, gekopieerd worden. **QUIET** betekent dat niet meegedeeld wordt welke files worden gekopieerd. **COPY** kan naar alle devices bestanden copiëren. Bijvoorbeeld, u wilt uw net afgewerkte bestand naar de printer sturen. Dit kan met het commando **COPY <bestandsnaam> TO prt:**

Verder kunt u gegevens sturen of halen van of naar SER: (seriële poort), PAR: (parallele poort), DFx: (alle diskdrives), DH0: (harddisk), RAM: (ramdisk), etcetera. Nog even een waarschuwing. Als er in de doeldirectory al een file staat met de naam die u opgegeven heeft, wordt hier zonder pardon overheen geschreven. Als laatste dan nog een opmerking, voor het copiëren van een diskette is het mogelijk het COPY-commando te gebruiken maar aan te raden is het niet. Het DISKCOPY commando is beter op deze taak berekend, bovendien zijn er verder nog genoeg P.D-kopieer programma's verkrijgbaar die hiervoor nog geschikter zijn.

RENAME

RENAME heeft tot taak, de naam van een bestand te veranderen. De syntax van dit programma is zeer eenvoudig:

RENAME <oude naam> <TO> <nieuwe naam>

Ook hier kan de naam bestaan uit een path, dit wil zeggen, een directory aanwijzing gescheiden door een '/' en vervolgens de naam van het te 'renamen' programma. Eventueel kan voor de directory aanwijzing nog een device naam als df0:, df1: etc. staan. Een tweede mogelijkheid is, is het verplaatsen van een bestand of directory binnen de directory-structuur. Hiermee kunt u een directory binnen een andere directory plaatsen, bijvoorbeeld u heeft weer de directory 'basicdemos'. Dit wilt u binnen de directory 'basicprogs' hebben. De opdracht

RENAME basicdemos TO basicprogs/basicdemos

verzorgt dit.

LIST

Toen wij dit commando voor de eerste keer zagen vroegen we ons echt af: 'waarvoor dient dit?' Je hebt toch een DIR commando? Heel snel zijn we er achtergekomen dat onze eerste gedachten fout waren. Oké, MS-DOS heeft het wel beter opgelost met de verschillende opties voor DIR maar toch blijft LIST een handig commando voor Amiga-DOS. LIST geeft meer informatie dan het DIR-commando. De format van LIST is: LIST <path><opties>. Zonder path (letterlijk vertaald, pad, weg; de directories waarbinnen het commando moet zoeken) neemt LIST aan dat de actuele directory bedoeld wordt. De output van LIST gebeurt volgens het formaat, programmanaam, grootte (in bloks) of directory vermelding (dir), flags, aanmaakdatum, -tijd. De flags staan altijd in de combinatie 'rwed' De letter staan voor:

r - READ, lezen is toegestaan

w - WRITE, schrijven is toegestaan.

e - EXECUTE, programma is uitvoerbaar

d - DELETE, programma mag worden gewist.

De flags r,w,e hebben nu nog geen functie, dat wil zeggen, AmigaDos trekt zich niets van de status van deze flags aan. De datum en de tijd geven aan wanneer het bestand voor het laatst 'geupdated' is. Ook bij LIST komen we weer diverse opties tegen. Deze zijn achtereenvolgens:

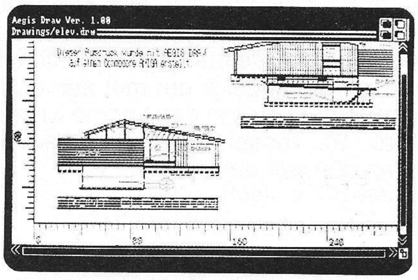
DIR: in welke directory gezocht moet worden.

PAT: aan welk patroon moet voldaan worden, ook hierbij komen jokers goed van pas.

KEYS: geeft aan vanaf welke track de file staat.

NODATES: onderdrukt de weergave van de datum.

TO: hierdoor kunt u de output van LIST naar ander device of file sturen. Verder bestaan nog de opties SINCE en UPTO. Hiermee kunt u de output van LIST vanaf of tot een bepaalde update datum regelen. Stel u wilt de directory entries sinds 19 november 1987 zien. U typt dan in: LIST SINCE 19-nov-87. De datumvermeldingen hoeven niet persé data te zijn. De Amiga kent ook nog de vermeldingen Today, Yesterday, en alle weekdagen. U kunt deze ook zonder moeite gebruiken. De regel 'LIST SINCE 19-nov-87 UPTO Yesterday' is dan ook een geldige invoerregel.



DISKCOPY

Een vorm van COPY, alleen deze copieert niet files maar een gehele schijf. De syntax van DISKCOPY:

DISKCOPY <FROM><drive>TO <drive> <NAME <name>>.

Bij DISKCOPY hoeft u de doeldiskette niet eerst te formatteren, dit wordt door het programma gedaan. Pas wel op welke schijf u gebruikt, alle gegevens die er op geschreven staan zijn onherroepelijk verloren. De optie FROM is optioneel, u mag het dus weglaten, TO is verplicht. Als u de kopie een andere naam wilt geven dan biedt het laatste deel van het commando hiertoe gelegenheid. Doormiddel van NAME <naam> geeft u de doeldiskette een andere naam. Als u met 1 diskdrive moet werken kan het een langdurige zaak worden, op zich een goede training voor het jongleren, maar na een aantal keren wisselen met schijven zult u toch echt schoon genoeg hebben van dit gedoe en schreeuwend van ellende naar de computerwinkel hollen om een tweede drive.

RELABEL

Met rename kunt u files een andere naam geven, RELABEL biedt deze mogelijkheid voor de naam van een diskette. Om de diskette in drive 1 de naam bootschijf te geven, typt u in: RELABEL df1: bootschijf. Het is mogelijk om spaties te gebruiken maar het is niet aan te raden.

INFO

Een instructie zonder parameters. Hoe wonderlijk! Toch is het een zeer handige instructie. Op geen andere manier dan met dit commando is het aantal vrije blokken op de schijf te controleren. De uitvoer van het commando begint met de aangesloten diskdrives, harddisks etcetera., de 'mounted disks' genaamd. Dan staat er vermeld wat voor devices er zijn aangesloten, hoe groot ze zijn, hoeveel blokken er gebruikt zijn, hoeveel er nog vrij zijn, het aantal errors (zeer rampzalige zaak, altijd voorkomen dus!) de status van het device (write of read) en als laatste, de naam. Onder 'Used' wordt het aantal bezette blokken vermeld. 'Free' geeft het aantal vrije blokken weer. 'Full' is de verhouding vrije blokken en bezette blokken (2 blokken = 1 KByte). Als tweede staat dan nog vermeld 'volumes available'. Dit geeft aan met welke diskettes de WorkBench is opgestart en welke diskettes voorhanden zijn zonder te wisselen.

INSTALL

Wie heeft er nog geen last gehad van virussen? Dan is het al vrij duidelijk waar dit commando voor dient. Een soort virus verwijderaar zouden we bijna zeggen. Helemaal onwaar is dit zeker niet. Het commando dient om een schijf opstartschijf te maken. De syntax van het commando luidt:

INSTALL <DRIVE> (drive)

DRIVE is optioneel. Na DRIVE dient dan wel het drivenummer te komen staan. Na een <RETURN> start het programma en zorgt ervoor dat er een nieuwe bootsector op de schijf wordt geschreven. Precies op de plaats waar de meeste virussen zitten!

TYPE

Dit commando 'typt' informatie op het beeldscherm. Voor degenen die MS-DOS kennen is dit een zeer bekend commando. Met dit commando kunt u de inhoud van een file, leesbaar of niet leesbaar, op het beeldscherm afbeelden. Bij het DIR commando hadden we ook al zoiets. (DIR opt i) De uitvoer stopt niet als het scherm vol is. Hier is wel een mogelijkheid voor. Druk een

toets en de uitvoer stopt tijdelijk. Met het indrukken van een willekeurige toets zal de uitvoer weer door gaan. De syntax:

TYPE <programmanaam> <opties>.

Ook bij TYPE is weer het gebruik van een 'redirection' teken mogelijk. Met TYPE >prt: ram:test.txt wordt de uitvoer van test.txt naar de printer gestuurd. Bij een redirection, of omleiding teken gaat het dus om het groter dan teken. Als u nu van een machinetaal programma een nette uitvoer wilt hebben kunt u de optie 'h' toevoegen. 'TYPE <naam> opt h' geeft een hexadecimale dump van een file. Helemaal rechts staan de leesbare tekens afgebeeld. Niet afdrukbare tekens worden vervangen door een punt. Verder kent TYPE dan nog de optie 'n'. De 'n' staat voor number. De uitvoer van elke regel van de file zal dan voorafgegaan worden door een regelnummer.

JOIN

Zoals het woord al zegt, voeg samen. De instructie kan maximaal 15 bestanden samenvoegen tot een nieuw bestand. De syntax luidt:

JOIN <bestanden (gescheiden door een komma)> AS <nieuw bestand>.

AS is hierbij een verplicht keyword. Een voorwaarde bij JOIN is, dat in de reeks bronbestanden elke filenaam maar 1 keer mag voorkomen. Ook bij JOIN is het gebruik van jokers weer toegestaan. JOIN kan als een soort TYPE dienen door de uitvoer om te leiden naar het beeldscherm met behulp van het '*' teken of om de uitvoer van meerdere bestanden achter elkaar naar de printer te sturen door na AS prt: te vermelden.

SEARCH

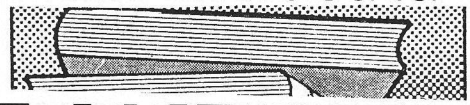
Tja, de vlag dekt de lading. Met dit commando kunt u files doorzoeken op een gegeven reeks tekens. Het commando luidt: SEARCH <FROM> <SEARCH> <ALL> <string> FROM, SEARCH en ALL zijn weer optioneel. Met FROM geeft u het door te zoeken path naar een directory of bestand aan. Met de toevoeging SEARCH geeft u aan dat het commando in de actuele directory moet zoeken, bijvoorbeeld; search "testnaam". Hierbij wordt in de actuele directory gezocht, door alle files heen, naar 'testnaam'. Als SEARCH iets vindt dan bestaat de uitvoer uit: de naam van de file gevolgd door het nummer en de tekst van regel waar de zoekstring instaat. Als laatste toevoeging is er nog ALL. All forceert SEARCH tot het doorzoeken van alle directories (kun je beter niet doen op een 30 MB harddisk). Wel even een waarschuwing, bij uitzondering maakt SEARCH wel onderscheid tussen kleine letters en hoofdletters in de zoekstring. Ook is bij SEARCH weer het gebruik van jokers in de door te zoeken files toegestaan. SEARCH #?word test, zoekt in alle files die eindigen op word, naar het woord 'test'.

TOT SLOT.

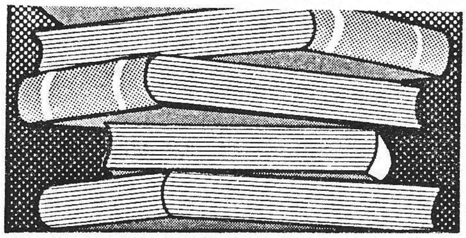
Wij hopen u met deze AmigaDos cursus van dienst te zijn geweest. Schrijven en blijven schrijven is ons motto. Waar is uw brief? Kom en klim in de pen. Laat ons weten wat u er van vindt. Tot de volgende keer dan maar weer, en computer ze!! Maak niet te veel brokken met de Amiga diskettes.

JOHAN & JOHAN.

Zoals u allemaal wel zult weten is de Amiga een zeer gecompliceerde computer. Vaak is hulp van een computer collega zeer welkom wanneer u problemen heeft. Maar die zal ook niet alles weten. Een goede oplossing is dan het aanschaffen van enige Amiga boeken. Welke u het beste kunt kopen moet u zelf maar uitkiezen. Keuze is er voldoende dat kunt u hieronder wel zien.



HET ZWAARDERE WERK



Om met de Amiga kunst en vliegwerk uit te kunnen halen moet u wel het een en ander in huis hebben. Natuurlijk zult u niet direct alles van de Amiga weten. De enige echte goede oplossing is dan het zelf aanschaffen van enige dikke boeken. Vervolgens moet u die boeken eens goed gaan bestuderen en dan zult u al veel minder problemen hebben bij het werken op en aan de Amiga.

De standaard 'PILLEN' van Sybex zijn in de Engelse taal geschreven, dus die zijn niet voor een ieder weggelegd. Databecker is een firma die al zeer lang bezig is met het oplossen van dit soort taalproblemen. Met de boeken van DATABECKER moet het programmeren in basic al helemaal geen problemen meer opleveren. Voor de taal C zult u toch nog wat meer dan boeken in huis moeten hebben. Alle Databecker boeken zijn van oorsprong geschreven in de Duitse taal en worden door Bruna in het Nederlands vertaald. En doordat deze boeken in het Nederlands zijn geschreven hoeft de Amiga gebruiker niet meer 3 talen te beheersen.

Wegwijs op de Amiga

Dit 238 pagina's tellende boekwerk is vooral voor de 1000 bezitter een leuke aanwinst. Wanneer u nog niet zo veel over de Amiga weet is dit een aanrader. De titel zegt het al, u wordt wegwijs op uw Amiga. In dit boek wordt als eerste uitgelegd hoe de Amiga geinstalleerd moet worden. Bijna alle hardware staat in het boek vermeld, zo wordt er precies verteld welke printers er door de Amiga ondersteund worden. De basiseigenschappen zoals, Intuition (de interface tussen mens en computer), Muis besturing, vensters, Amiga opstarten, Preferences, het Noteblok, de Klok, Basic inleiding, de vuilnisbak van de Amiga (trashcan), en nog veel meer wordt in dit boek besproken. Te veel om op te noemen dus. Zo staan er achter in dit boekje 7 appendices over storingen, het toetsenbord, icons, verklarende woordenlijst, icon editor, technische gegevens en een trefwoorden register. Zeker voor de beginner een beste aanschaf. Dit boekje kost *f* 59,90 en is verkrijgbaar onder ISBN nummer: 90 229 3413 6.

Amiga tips en truuks

Dit 291 pagina's tellende boekwerk is direct al een stuk interessanter voor Amiga gebruikers. Of u nu een 500,1000 of 2000 in huis heeft staan het maakt niets uit. Dit boek start met de 23 meest gevraagde problemen met de Amiga. Hoe krijg ik wat op de printer, het aanmaken van een ramdisk, het kopieren van een programma met maar een drive, wat is een startup-sequence en wat kan ik er mee, kan de Amiga ook spreken en zo ja hoe. Dit zijn een paar van die veel gevraagde vragen. In hoofdstuk 2 wordt u met basic op weg geholpen. Diverse routines helpen u Amiga Basic te leren begrijpen. Persoonlijk werken wij liever in C dan in Amiga Basic omdat deze zo afschuwelijk traag is! En in hoofdstuk 3 wordt het, voor de aspirant C FREAKS onder ons, echt interessant. In dit hoofdstuk staan voorbeelden die u met behulp van een goede compiler moet compileren. U wordt op weg geholpen bij het maken van eigen Screens, windows en diverse andere voorbeelden. natuurlijk is dit boek geen C leerboek maar handig is het zeker wel. En laten we nu eerlijk zijn daar zitten alle C beginners op te wachten. Zeker gezien het aantal vragen over C. Door het goed bestuderen van dit hoofdstuk zal ook uw 'C lampje' in de bovenkamer gaan branden. Wanneer u als aspirant C freak wat wilt in de taal C zult u direct naar dit fraaie hulpmiddel moeten grijpen. Wij hebben persoonlijk alle programma's zitten intikken, en we moeten wel even vertellen dat wij een aantal problemen hadden bij het gebruik van de MANX Aztek C compiler. Ook met de C compiler van Lattice, waarmee de voorbeelden moesten werken, kregen wij sommige programma's niet aan de praat. Gelukkig kunt u bij Bruna na het aanschaffen van de boeken rekenen op een goede ondersteuning. Stuur maar een lege diskette op, dan krijgt u de goede C file's toegestuurd. Op deze manier kunt u reken op een goede ondersteuning. Dus werkt u met de MANX Aztek C compiler dan dient u direct na aanschaf een diskette naar BRUNA op te sturen (het adres staat voorin het boek). Vergeet niet een copy van de aankoopbon bij de diskette op te sturen. De werkende programma's voor de MANX compiler staan ook op de diskette die u terug krijgt. En wel in source vorm! Wanneer U deze diskette ontvangen heeft dient u alles met de +L optie te compileren, bij het linken dient u de optie -lc32 te gebruiken. Probeer dit ook eens bij de Sybex voorbeelden uit de standaard PILLEN. En u zult zien dat die vaak wel werken met deze opties. Nogmaals moeten wij er op wijzen dat dit geen leerboek voor de taal basic en/of C is. Daar zult u of het basic boek, welke hieronder wordt besproken, moeten aanschaffen. Tips en truuks kost *f* 69,90 en is verkrijgbaar onder ISBN nr: 90 229 3442 X

Het Amiga Basic boek

Het Amiga basic boek is een dikke pil. Met zijn 576 pagina's tellende omvang begrijpt u wel dat het hier om een soort leerboek gaat. Dus bent u of gaat u wat in basic ondernemen is dit

boek gewoonweg een MUST. Zo worden er in deze pil een aantal goede bruikbare programma's ontworpen. Dit boekwerk kopen is voor elke basic programmeur een aanrader. Het is een handige tool bij het ontwikkelen van diverse programma's. Ook het werken met gegevens bestanden is ruim aan bod. Natuurlijk is het werken in de grafische mode niet vergeten. Al lerende leert u uw Amiga kennen. En vergeet niet dat dit een heleboel nuttige programma's kan opleveren. Wie wil er nu niet met Bobs, Sprites, Screens, Windows kunnen werken. En daar is de Amiga nu juist zo goed in. Wanneer u daar GEEN gebruik van maakt had u net zo goed op de C64 kunnen blijven werken! Een spel spelen kunnen we allemaal wel, maar hoe zit dat met het programmeren. Wanneer u de programma's netjes heeft zitten intikken kunt u deze het beste met een goede basic compiler compileren. Dit boek kost *f* 59,90; ISBN nr: 90 229 3418 7.

Het Supergrafiekboek

Dit 424 pagina's tellende boekwerk is een must voor alle demo programmeurs in ons land. Natuurlijk ook alle spelprogrammeurs dienen dit boekwerk in huis te hebben. Er staat te veel om op te noemen in dit boek maar wij maken een kleine selectie. De grafische instructies van AmigaBasic, laden en opslaan van afbeeldingen in ILBM formaat, sprites ontwerpen, het werken met bobs en animatie, een CAD programma met 1024 * 1024 resolutie, het gebruiken van de speciale chips van de Amiga, grafische programmeren vanuit C, programmering van Copper en Blitter en als slot een complete beschrijving van het grafische systeem van de Amiga. Zo worden hier de View, ViewPort, RastPort, en de opbouw van de Bitmaps, Screens en Windows uitgelegd. Zonder verder te schrijven kunne wij u melden dat dit nummer 2 is van onze prive collectie! nummer een is het volgende boek wat we gaan bekijken. Het supergrafiek boek is geschreven door Jennrich, Trapp en Weltner en kost F 79,90 het is verkrijgbaar onder ISBN nr:90 229 3470 5

AMIGA INTERN.

Dit boek bevat 392 pagina's met waanzinnige tekst en uitleg over van alles en nog wat. Veelal technische info over de aansluitingen en de Interne Amiga. Zo worden alle poorten van de Amiga besproken. Ook de 68000-processor van Motorola (ongetwijfeld een van de krachtigste 16-bits microprocessoren) komt aan bod. Als u nog niet wist wie Agnus, Denise en Paula zijn dan kunt u dat in het boek lezen. Maar voordat die chips worden besproken kunt u eerst doormiddel van een blokschema de plaats van de custom chips bekijken. Ook wordt er in dit hoofdstuk gesproken over de 8520 CIA (de 68000 processor heeft via data-en adreslijnen direct toegang tot de beide CIA's en de kickstartrom), PAL-norm, chip-ram, DMA-controller en nog veel meer. Voor de gebruiker die assembly wil gaan programmeren is dit boek ook vrij handig. Hier staan namelijk de grondbeginselen en een aantal voorbeelden zoals: interrupts, de coprocessor Copper, Playfields, Sprites, de Blitter, geluid, de uitlezing van een Muis/joystick/paddles, reset beveiligingen en noem maar op. Voor de C gebruiker is dit boek ook vrij handig. Wie met succes wil programmeren zal in grote trekken het besturingsysteem moeten kennen. Dat wordt in het hoofdstuk EXEC uitgelegd. In dat hoofdstuk staat bijvoorbeeld de bewerking van lijsten met een Exec-routine, het bewerken van bibliotheken, multitasking, wachten op signalen, boodschappen, trapinstructies enz enz. AMIGA INTERN kost *f* 79,90 en is verkrijgbaar onder ISBN nr: 90229 3473 X

Amiga DOS

Dit zo rond de 300 pagina's tellende boek is iets wat u niet kunt missen bij uw boekencollectie. In dit boek worden alle DOS-commando's behandeld zoals: Format, Copy, Dir, CD, Makedir, Ed, Type, enz. Deze commando's geven u de mogelijkheid om schijven te formatteren, files te copieeren, met sub-directories werken of tekstfiles te lezen. Het is ook mogelijk vanuit de Dos met multitasking te werken, bijv. Terwijl u een disk aan het formateren bent kijkt u de tekst nog even na die u op diezelfde disk wilt wegschrijven. Als u dit boek heeft doorgewerkt zult u verder niet zoveel problemen met het DOS gebeuren hebben en er dus aardig goed mee om kunnen gaan, want nu heeft u teminste geen last meer met het aanmaken van een ram-disk (dat voor de meeste mensen het grootste probleem is), zodat u even een directory van een andere schijf kunt trekken. Het verwisselen van directories zal dan dus ook geen problemen meer opleveren. Het leuke van dit boek is dat het met de tijd meegaat, want de nieuwe commando's van de Workbench 1.3 staan in het boek vermeld en worden tevens besproken. Daar komt dan wel bij dat de Workbench 1.3 nog niet officieel is uitgegeven dus 'het naadje van de kous' zal dit boek dan ook niet kunnen beschrijven. Verder zit dit boek boordevol met handige tips en trucs die natuurlijk betrekking hebben op het AmigaDOS. Dit handige boek kost *f* 59.90 en is te bestellen onder ISBN nr: 90 229 3478 0

AMIGA 3D

Voor de 3D fanaten onder ons is dit boek met zijn 169 tellende dikke omvang een echte aanrader. Wel moeten wij even vermelden dat u zeker niet in de laatste zin stevig in de 'wiskunde schoenen' moet staan. Wiskunde in uw vakkenpakket is dus een pré. Na het doornemen van dit boek weet u bijna alles over 3D op de Amiga. Het boek wordt geleverd met een 3.5 inch diskette met daarop de in het boek beschreven programma's. In het boek wordt de ray tracing-methode principes besproken. De driedimentionale Bobs die met behulp van een handige editor kunnen worden ingevoerd zijn heel mooi om te zien of te gebruiken in uw eigen programmatuur. Met zijn technische informatie is dit boek zeker niet voor de beginner weggelegd. Bent u eenmaal op het Amiga pad gekomen, zult u zeker een keer aan dit boek toekomen. De Amiga is immers een pure grafische en animatie computer.

Tot slot

Tot slot kunnen wij zeggen dat wij zelf heel tevreden zijn met de boeken van DATABECKER. Natuurlijk zijn er ook nog andere op de markt verkrijgbaar, maar vaak niet in de Nederlandse taal. Wij Nederlanders staan dan wel bekend om onze flexibiliteit wat taal gebruik betreft maar natuurlijk zijn er personen die het daar niet geheel mee eens zijn. En juist voor die mensen is de gehele DATABECKER reeks zo aantrekkelijk. Bovendien verdient deze firma dat haar boeken worden gelezen. Wanneer we kijken naar de informatie die in de boeken staat en vergelijken wat dat voor een hoeveelheid is mogen we niet klagen over de prijskaartjes die aan deze boeken hangen.

NewInput

De AmigaBasic die je bij De Amiga geleverd krijgt heeft zoals De meeste basics geen fantastisch goede INPUT routine. De programmeur kan er bijvoorbeeld niet van op aan of je in het vak blijft waar de programmeur wil dat je blijft, zodat je de andere scherminformatie niet kan verminken. je kunt zelfs nog geen limiet meegeven wat betreft het aantal tekens en dat soort dingen. Nu is de enige oplossing om dit probleem te laten verdwijnen je eigen INPUT routine schrijven.
Het programma bestaat uit 2 subs: één voor de wat speciale instellingen, die je niet al te gauw zult gebruiken en dan de eigenlijke INPUT-routine. Deze routines heten: Manier en Invoer.



Manier

Manier is de SUB die voor de eigenlijke routine INVOER aangeroepen moet worden, om enkele speciale instellingen te doen. dit is gedaan om de gebruiker van dit programma anders moet hij bijvoorbeeld intikken: 2,33,34,1,"blablabla",34,2,1,9,0,2, en dat lijkt me niet zo super. Natuurlijk zitten de meest gebruikte functies gewoon in het commando INVOER verwerkt.

het commando Manier is al volgt samengesteld:
Type, Beveiliging, Schoon, Return
De opties zijn:

Type:
De gebruiker kan uit 5 verschillende invoersoorten kiezen en wel:
0: Alle karakters/toetscombinaties toegestaan (De normale stand)
1: "0123456789.-/*^()/+=" Alle rekenkundige operatoren dus
2: Aleen letters en leestekens
3: "jJnNyY" dit lijkt me duidelijk
4: "0123456789." Alleen voor numerieke INVOER

Beveiliging:
Het standaard Type Beveiliging, namelijk als De gebruiker zijn code intypt Dan verschijnt daar een "-" voor. Niemand kan dus even meegenieten. [0]=Beveiliging UIT en [1]=Beveiliging AAN

Schoon:
Als de gebruiker het gebied waar de INVOER moet komen staan, Schoon wil hebben is dit optie [1]0! wil De gebruiker dat dit niet gebeurt Dan gebruikt hij optie [0]0!

Return:
Soms wil de gebruiker dat iemand niet gelijk op RETURN kan drukken, als hij nog niks heeft ingevoerd. Dus [0]= verboden en [1]= toegestaan.

INVOER

INVOER is De eigenlijk routine die De INVOER behandeld en bestuurt. Het commando is:

INVOER [imiet],[voorgrond],[achtergrond], [text$], [xpos],[ypos]

[limiet]: het aantal tekens dat ten hoogste geaccepteerd zal worden.
[voorgrond]: De voorgrondkleur
[achtergrond]: De achtergrondkleur
[text$]: een string uitdrukking dus bv a$+CHR$(13)+"Cinfo"
[xpos]: De x-positie van de te plaatsen text
[ypos]: De y-positie van de te plaatsen text

Opmerking: is xpos -1, dan zal de inputroutine de laatste x en y positie gebruiken van de cursor.

```
' Amiga Basic V1.2 - 80 Koloms/Commodore
Info Nederland [RETURN]
' Een echte Input routine met alle moge
lijkheden zoals [RETURN]
' delete/backspace/cursor/limiet/beveili
ging/verschillende types enz. [RETURN]
' die beter resultaat biedt bij bijvoor
beeld de AC/BASIC COMPILER. [RETURN]
' R.Sloot 01659-3686 [RETURN]

' *** UITLEG *** [RETURN]
DIM SHARED cursorbeeld%(18) 'Dit moet in
het programma staan [RETURN]
CLS:LOCATE 2,1:PRINT "Deze veel betere
input-routine bestaat uit 2 Sub's" [RETURN]
PRINT "en u kunt de commando's dan ook mak
kelijk aanroepen met bv INVOER enz." [RETURN]
FOR I=0 TO 4000:NEXT:PRINT:PRINT "\ Over
zicht Mogelijkheden \" [RETURN]
PRINT " format: INVOER limiet,voorgrondkleur
,achtergrondkleur,tekststring,x,y" [RETURN]
```

```
PRINT " speciale instellingen: MANIER type,
beveiliging,schoonmaken,return" [RETURN]
PRINT:PRINT "\ Uitleg Mogelijkheden \":
PRINT:PRINT " (Invoer)" [RETURN]
PRINT "-  : stelt het aantal tekens
voor dat geaccepteerd moet worden" [RETURN]
PRINT "-
/Achtergrondkleur : kleu
ren van letters/letterveld" [RETURN]
PRINT "- ,y : Positie van tekst
string" [RETURN]
PRINT "  is =-1 dan wordt de string ach
ter de laatste cursorpositie geprint" [RETURN]
PRINT:PRINT " (Manier)" [RETURN]
PRINT "- ype type 0 : alles toegestaan"
[RETURN]
PRINT SPACE$(9)"type 1 : 0123456789.-/*^(
)/+=" [RETURN]
PRINT SPACE$(9)"type 2 : alleen letters
en leestekens" [RETURN]
PRINT SPACE$(9)"type 3 : jJnNyY" [RETURN]
PRINT SPACE$(9)"type 4 : 0123456789."
[RETURN]
PRINT "- ging : 0=uit 1=aan"
[RETURN]
PRINT "- hoonmaken : 0=uit 1=aan :
wissen van tekstgedeelte" [RETURN]
PRINT "-
    : 0=return niet toe
gestaan als niks is ingevuld" [RETURN]
PRINT SPACE$(16)": 1=gelijk op return
drukken is toegestaan" [RETURN]
WHILE INKEY$="" [RETURN]
WEND [RETURN]
CLS:invoer 10,1,2,"Testen maar! ",10,5
[RETURN]
COLOR 1,0:LOCATE 8,10:PRINT "U heeft in
gevoerd: "in$ [RETURN]
' Na aanroep routine is de invoer opge
slagen in in$ of in(numeriek) [RETURN]
LOCATE 9,10:PRINT "eventuele numerieke
waarde: "in [RETURN]
FOR I=0 TO 25000:NEXT [RETURN]
END [RETURN]
SUB Manier(g%,g1%,g2%,g3%) STATIC [RETURN]
SHARED type%,Beveiliging%,Schoon%,Ret%,
Speciaal% [RETURN]
Speciaal%=1:type%=g%:Beveiliging%=g1%:
Schoon%=g2%:Ret%=g3% [RETURN]
END SUB [RETURN]

SUB invoer(limiet%,VoorGrond%,AchterGrond
%,Text$,xpos%,ypos%) STATIC [RETURN]
SHARED in$,Speciaal%,type%,Beveiliging%,
Schoon%,Ret%,in [RETURN]
'Initialiseer variabelen/voorbereidingen
[RETURN]
IF xpos%=-1 THEN xpos%=POS(0):ypos%=
CSRLIN [RETURN]
LOCATE ypos%,xpos%:PRINT Text$;:inp$="":
csr%=0:insert=0 [RETURN]
COLOR VoorGrond%,AchterGrond% [RETURN]
IF Speciaal%=0 THEN Ret%=1:Beveiliging%=
0:type%=0:Schoon%=1   [RETURN]
IF Schoon%=1 THEN PRINT SPACE$(limiet%);:
LOCATE CSRLIN,POS(0)-limiet% [RETURN]
Speciaal%=0:COLOR VoorGrond%,AchterGrond%
[RETURN]
xpo=(POS(0)-1)*8:ypo=(CSRLIN-1)*8 [RETURN]
LINE (xpo,ypo)-STEP (1,7),VoorGrond%,b
[RETURN]
GET (xpo,ypo)-(xpo+1,ypo+7),cursorbeeld%
[RETURN]

IF type%=0 THEN [RETURN]
st=1:lo%=32:hi%=255:l1%=lo:l2%=hi [RETURN]
ESEIF type%=1 THEN [RETURN]
st=0:st$="0123456789-.*/^()+="   [RETURN]
ELSEIF type%=2 THEN [RETURN]
st=1:lo%=32:hi%=47:l1%=48:l2%=255 [RETURN]
ELSEIF type%=3 THEN [RETURN]
st=0:st$="jnJNyY" [RETURN]
ELSEIF type%=4 THEN [RETURN]
st=0:st$="0123456789." [RETURN]
END IF [RETURN]

' Begin eigenlijke routine [RETURN]

Tsloop: [RETURN]
key$="" [RETURN]
WHILE key$="" [RETURN]
key$=INKEY$ [RETURN]
WEND [RETURN]
lo$=key$:po%=ASC(lo$):key$=UCASE$(key$):
key%=ASC(key$) [RETURN]
xpo=(POS(0)-1)*8:ypo=(CSRLIN-1)*8 [RETURN]
IF key%=13 THEN [RETURN]
IF LEN(in$)=0 AND Ret%=0 THEN BEEP:GOTO
Tsloop [RETURN]
PUT (xpo,ypo),cursorbeeld%,XOR:in=VAL
(in$) [RETURN]
EXIT SUB [RETURN]
ELSEIF key%=8 THEN  [RETURN]
IF csr%=0 THEN Tsloop [RETURN]
PUT (xpo,ypo),cursorbeeld%,XOR [RETURN]
buf$=LEFT$(in$,csr%-1):rest$=MID$(in$,
csr%+1,LEN(in$)):in$=buf$+rest$ [RETURN]
ev=POS(0)-1:LOCATE CSRLIN,ev:PRINT
rest$+" "; [RETURN]
LOCATE CSRLIN,ev [RETURN]
PUT (xpo-8,ypo),cursorbeeld%,XOR [RETURN]
csr%=csr%-1:GOTO Tsloop [RETURN]
ELSEIF key%=127 THEN [RETURN]
IF csr%=limiet% THEN Tsloop [RETURN]
PUT (xpo,ypo),cursorbeeld%,XOR   [RETURN]
buf$=LEFT$(in$,csr%):rest$=MID$(in$,
csr%+2,LEN(in$)):in$=buf$+rest$ [RETURN]
ev=POS(0):LOCATE CSRLIN,ev:PRINT rest$
+" "; [RETURN]
LOCATE CSRLIN,ev [RETURN]
PUT (xpo,ypo),cursorbeeld%,XOR [RETURN]
GOTO Tsloop [RETURN]
ELSEIF key%=31 THEN [RETURN]
IF csr%=0 THEN Tsloop [RETURN]
insert=1:sort=-8:LOCATE CSRLIN,
POS(0)-1 [RETURN]
csr%=csr%-1:GOSUB LineSetUnset:
GOTO Tsloop [RETURN]
```

```
ELSEIF key%=30 THEN [RETURN]
IF csr%=limiet% OR csr%=LEN(in$) THEN
insert=0:GOTO Tsloop [RETURN]
insert=1:sort=8:LOCATE CSRLIN,POS(0)+1
[RETURN]
csr%=csr%+1:GOSUB LineSetUnset:
GOTO Tsloop [RETURN]
END IF [RETURN]
IF st=1 THEN [RETURN]
IF (po%% OR po%hi%) AND (po%1% OR
po%l2%) THEN Tsloop [RETURN]
ELSEIF st=0 THEN [RETURN]
IF INSTR(st$,lo$)=0 THEN Tsloop [RETURN]
END IF [RETURN]

IF csr%=limiet% THEN Tsloop [RETURN]
PUT (xpo,ypo),cursorbeeld%,XOR [RETURN]
IF insert=0 THEN [RETURN]
in$=in$+lo$ [RETURN]
IF Beveiliging%=1 THEN PRINT "-";
ELSE PRINT lo$; [RETURN]
ELSE [RETURN]
buf$=MID$(in$,csr%+1,LEN(in$)):in$=
LEFT$(in$,csr%) [RETURN]
voeg$=lo$+buf$ [RETURN]
IF LEN(in$+voeg$)limiet% THEN [RETURN]
in$=in$+buf$ [RETURN]
PUT (xpo,ypo),cursorbeeld%,XOR [RETURN]
GOTO Tsloop [RETURN]
END IF [RETURN]
in$=in$+voeg$:ev=POS(0)+1 [RETURN]
IF Beveiliging%=1 THEN [RETURN]
PRINT STRING$(LEN(voeg$),"-");:LOCATE
CSRLIN,ev [RETURN]
ELSEIF Beveiliging%=0 THEN [RETURN]
PRINT voeg$;:LOCATE CSRLIN,ev [RETURN]
END IF [RETURN]
END IF [RETURN]
PUT (xpo+8,ypo),cursorbeeld%,XOR [RETURN]
csr%=csr%+1:u=CSRLIN:o=POS(0) [RETURN]
GOTO Tsloop [RETURN]

LineSetUnset: [RETURN]
PUT (xpo,ypo),cursorbeeld%,XOR [RETURN]
PUT (xpo+sort,ypo),cursorbeeld%,XOR [RETURN]
RETURN [RETURN]

END SUB [RETURN]
```

